Titre: Title:	An empirical study of expert recommendations for the algorithm design of an intelligent study guide
Auteur: Author:	Lei Ma
Date:	2006
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Ma, L. (2006). An empirical study of expert recommendations for the algorithm design of an intelligent study guide [Mémoire de maîtrise, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/7895/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/7895/
Directeurs de recherche: Advisors:	
Programme: Program:	Non spécifié

UNIVERSITÉ DE MONTRÉAL

AN EMPIRICAL STUDY OF EXPERT RECOMMENDATIONS FOR THE ALGORITHM DESIGN OF AN INTELLIGENT STUDY GUIDE

LEI MA

DÉPARTEMENT DE GÉNIE INFORMATIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCE APPLIQUÉES
(GÉNIE INFORMATIQUE)
AOÛT 2006

Your file Votre référence
ISBN: 978-0-494-19314-3
Our file Notre référence
ISBN: 978-0-494-19314-3

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire institulé:

AN EMPIRICAL STUDY OF EXPERT RECOMMENDATIONS FOR THE ALGORITHM DESIGN OF AN INTELLIGENT STUDY GUIDE

Présenté par: MA, Lei

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

a été dûment accepté le jury d'examen constitué de:

M.BOUDREAULT, Yves, Ph.D., président

M.DESMARAIS, Michel., Ph.D., membre et directeur de recherche

M.ROBILLARD, Pierre-N., Ph.D., membre

ACKNOWLEDGEMENTS

I would like to thank, first and foremost, my supervisor Dr. Michel C. Desmarais for his helpful discussions, ideas, and invaluable guidance throughout the duration of this work. His enthusiasm, dedication, and expertise were immersed into any little progress of this study. For all of these, I will always be grateful in my lifetime.

I would like to thank my colleague, Mihaela Dulipovici, for her constant help and advice for my research, and to thank other colleagues: Xiaoming Pu, Shunkai Fu, Peyman Meshkinfam, Alejandro Villarreal Morales and my friend Lan Kong for their friendship, support, and suggestions for my work.

Moreover, I would like to thank my family and friends for their encouragement and support.

RÉSUMÉ

Les guides d'études visent à fournir des recommandations personnalisées pour aider l'apprenant à se retrouver dans un grand volume de contenu didactique, par exemple des répertoires d'objets d'apprentissage ou encore la matière d'une année ou d'un semestre d'un cours. Nous étudions les exigences auquelles on peut s'attendre de tels guides.

Nous avons demandé à huit instructeurs d'effectuer des recommandations d'étude, à partir des résultats de six étudiants à un test, et recuilli leurs recommandations ainsi que leurs impressions avec un questionnaire. Cette expérimentation nous permet de tirer quelle information ils utilisent, quel degré de confiance ils ont en leurs recommandations et de mesurer l'accord entre eux quant aux recommandations. En plus de nous servir de base à l'établissement d'un algorithme de recommandation, les résultats indiquent que les huit instructeurs ont un degré relativemen élevé d'accord quant à leurs recommandations.

Dans une seconde expérience, nous tentons, avec un algorithme simple, de reproduire les recommandations d'un instructeur avec un échantillon de 51 cas de recommandations. Les résultats démontrent que, malgré le fait que les instructeurs semblent utiliser de l'information tirée des réponses détaillées des étudiants et auquel un guide informatisé n'a généralemet pas accès, l'algorithme réussit néanmoins à répliquer relativement fidèlement les recommandations d'un instructeur.

ABSTRACT

Personalized service for web-based learning has recently received considerable attention because of different needs among users. Most recommendation systems consider learner/user preferences, interests, or browsing behaviors when analyzing learner behaviors for personalized services. However, these systems neglect the importance of learner knowledge level for implementing personalized mechanisms. A major challenge for web-based educational systems is to provide students with personalized learning instructions, such as the most suitable pedagogical recommendations that best match their knowledge level.

The work presented in this thesis is a part of a larger ongoing project: the design and implementation of a web-based Adaptive Educational System, the Poly Study Guide, which can guide student knowledge remediation by making personalized assessment-driven recommendation. A study recommendation module in this study guide can be guided by an item inference engine, such as Partial Order Knowledge Structures (POKS)(Desmarais, Meshkinfam and Gagnon,2006). Two objectives of this thesis are: to investigate the requirements for the recommendation module in the intelligent study guide, and to devise an appropriate algorithm that can grant the study guide the ability to diagnose knowledge states and make individual study recommendation.

In order to investigate the requirements for the recommendation module, we realized an experiment with eight experienced instructors to investigate the process of one-on-one tutoring. More specifically, we collected information in this experiment about professional knowledge diagnosis and study plan recommendations for each individual student, and examined the agreement among recommendations from

different instructors in order to determine what desired recommendation results a study guide should deliver. Besides, we also attempted to determine the value of detailed answers for improving recommendations.

Both quantitative and qualitative approaches were employed for analysis and interpretation of the findings in this experiment. Some major findings are briefly summarized here. The agreement among recommendations from eight instructors is substantial. Thus, we can consider those results as expert recommendations and emulate them in the recommendation module of our proposed study guide. In addition, all instructors made corrections in their recommendations after they evaluated students' complete answer sheets, especially in the case of making recommendations for students in medium knowledge level. Besides, in their responses to a questionnaire, the instructors revealed strongly positive perceptions toward the value of the complete answer sheet. Thus, a computerized study guide may not make as good recommendations as a professional human instructor unless it could analyze the answers.

We devised a q-matrix and a relatively simple but effective algorithm in the recommendation module that could emulate the expert recommendations collected in the experiment. Furthermore, a simulation test was performed to validate the effectiveness of the recommendation module in our intelligent study guide. We compared the recommendations from this intelligent study guide with those from the experts.

The results of this simulation test show that the accuracy of recommendations from our program is superior to random recommendations and it increases gradually when more items are administered. When the responses to all the eight sub-question items are given, the accuracy of this recommendation module reach almost 90%. These

findings can basically confirm the effectiveness of this recommendation algorithm. Besides, the results of the simulation test also confirm that more accurate the item assessment is, more accurate the recommendation is. Thus, the accuracy of recommendation is also dependent upon the effectiveness of the item assessment engine. Undoubtedly, a good item assessment engine, which can efficiently infer accurate performance of a student, is the solid basis for making proper recommendation.

Keywords: personalized assessment-driven recommendation, intelligent study guide, web-based Adaptive educational system, q-matrix, adaptive assessment, POKS

CONDENSÉ EN FRANÇAIS

Introduction

Les logiciels de formation, qu'ils soient à distance, basés sur le Web ou diffusés sur médium CD, gagnent en popularité et remplacent progressivement les instruments traditionnels de formation comme les livres académiques. Cependant, la majorité de ces applications n'exploitent pas à fond toutes les possibilités d'interactivité et d'adaptabilité. Elles demeurent encore loin de fournir un contenu personnalisé en fonction de l'individu, comme on pourrait s'attendre d'un tuteur humain, par exemple.

Un des grands défis des logiciels de formation est celui de s'adapter au style d'apprentissage, aux intérêts et, notamment, de s'adapter au niveau des compétences acquises de l'apprenant. La personnalisation du cheminement d'apprentissage vis-à-vis du niveau de connaissance de l'apprenant est le sujet de ce mémoire. Nous visons plus spécifiquement à établir les exigences d'un guide d'étude personnalisé. Les travaux de ce mémoire s'inscrivent dans le cadre d'un projet de plus grande envergure visant à développer un guide d'étude pour les étudiants de Poly, auquel nous référerons ici par l'expression "Guide d'étude Poly" (Poly Study Guide).

Ce mémoire comporte deux objectifs spécifiques. Le premier est de contribuer à établir les exigences quant aux recommandations faites par un guide d'étude; le second est de concevoir un algorithme qui effectue des recommandations personnalisées basées sur l'état de connaissance de l'apprenant et de valider l'exactitude de ces recommandations. Ces objectifs constituent des éléments de base du Guide d'étude Poly.

Le premier objectif est abordé par une expérimentation où l'on observe des instructeurs et experts du domaine fournir des recommandations d'étude à des étudiants en se basant sur les résultats d'un test. On cherche ainsi à savoir quelles recommandations ils formulent en fonction des résultats, l'information qu'ils utilisent, leur niveau de confiance dans leurs recommandations, etc. Le second objectif fait plutôt l'objet d'une simulation où l'on tente de reproduire les résultats des recommandations par un modèle algorithmique. Ces deux expérimentations seront décrites plus loin.

Recommandations d'étude et engins d'inférence de la connaissance

Les recommandations d'études, c'est-à-dire les sujets sur lesquels l'apprenant doit se concentrer, constituent l'élément central d'un guide d'étude. Khuwaja, et al. (1996) définissent le but d'un guide d'étude intelligent comme celui de fournir des directives pédagogiques basées sur l'évaluation de l'état de connaissance, comme les concepts à approfondir, afin d'acquérir le niveau de maîtrise désiré quant à un ensemble de sujets ou à un domaine d'expertise. Le Alberta Research Council (1995) rapporte "qu'afin de concevoir des directives individualisées, il est premièrement nécessaire d'établir le niveau de compréhension que l'étudiant a d'un sujet" (traduction libre).

Généralement, les recommandations personnalisées d'un guide d'étude s'appuient sur une évaluation précise (*fine-grained*) des connaissances. Plusieurs approches probabilistes ont été développées pour fournir une telle évaluation comme les réseaux bayésiens, les espaces de connaissance (Falmagne, Koppen et al., 1990), la Théorie des Réponses aux Items (TRI) et ses dérivés comme le "Fuzzy Item Response Theory (FIRT)" (Chen, Duh et al., 2004), etc.

L'approche POKS, *Partial Order Knowledge Structures*, (Desmarais, Meshkinfam, Gagnon, 2006) est basée sur la théorie des espaces de connaissances. Il s'agit d'une approche bayésienne qui peut être paramétrisée à partir de données et évite le lourd effort d'ingénierie de la connaissance qui caractérise les approches basées sur les réseaux bayésiens. Elle crée des structures d'inférences entre les questions (items) elles-mêmes. Cette particularité lui permet d'aborder le problème de la construction de la structure des questions avec une approche d'apprentissage automatique, sans intervention humaine, d'où le gain en ingénierie de connaissance. Le résultat est donc celui de prédire les réponses à toutes les questions à partir d'un sous-ensemble, les questions observées. On envisage que c'est avec cet engin que le Guide d'étude Poly effectuera un diagnostic de connaissance.

Une fois la probabilité de succès aux questions déterminée par l'engin d'inférence, l'approche POKS nécessite une seconde étape où l'on doit établir la maîtrise des concepts à partir de ces probabilités de succès aux questions. Un modèle relativement simple mais efficace d'effectuer ce lien entre les réponses et les concepts et de produire un diagnostic précis des compétences acquises est le modèle Q-matrice (Tatsuoka, 1983). Ce modèle consiste à créer une matrice questions-concepts qui détermine quels concepts sont associés à quelles questions. Bien que la définition d'un modèle pour un domaine donné implique un travail de réflexion qui ne peut être facilement automatisé, il a l'avantage d'être facile à comprendre pour des instructeurs qui font régulièrement une tâche analogue lorsqu'ils construisent un examen, par exemple. L'algorithme que nous proposons pour le Guide d'étude Poly se repose donc sur l'approche Q-matrice. Il est décrit plus loin.

Expérience avec des instructeurs

La première expérimentation vise à éclairer quelles sont les exigences d'un guide quant aux recommandations. L'objectif consiste à s'inspirer de ce que font des instructeurs d'expérience pour déterminer ces exigences. Elle fut réalisée avec huit instructeurs qui ont eu à effectuer une tâche de recommandation d'étude pour six étudiants avec des résultats que ces étudiants ont obtenu à un test. Chaque instructeur devait indiquer, pour chacun des 19 sujets (*concepts* dans la Q-matrice) ceux qui devaient *à étudier en priorité*, *à réviser* et *bien maîtrisés*. Nous mesurons l'exactitude des recommandations en les comparant à celles faites par un instructeur. Les recommandations sont de trois types : « maîtrisé » *(mastered),* « réviser » (*review*) et « étudier » (*focus*). Une recommandation est correcte si elle correspond à celle de l'expert.

Nous avons recueilli et analysé leurs recommandations qui déterminent en quelque sorte la norme qu'un guide devrait respecter et que nous viserons à répliquer avec l'algorithme proposé. L'expérience nous permet aussi de déterminer quelle information est nécessaire pour poser un diagnostic précis des connaissances acquises. Elle permet notamment d'évaluer la valeur, aux yeux des instructeurs, de l'accès aux réponses détaillées par rapport à l'accès uniquement aux résultats du test, sachant que nous envisageons que le Guide d'étude n'aura accès qu'aux résultats.

L'analyse des données de l'expérience se base sur des approches à la fois qualitatives et quantitatives. Les principales conclusions de l'analyse sont les suivantes :

Analyse des recommandations effectuées :

1. L'accord entre les huit instructeurs quant à leurs recommandations pour les six étudiants est relativement grand, avec un score Kappa moyen qui se situe aux environs de 0.70. Ce résultat est important puisqu'il indique une certaine convergence des recommandations et confirment qu'on peut présumer que ce sont des recommandations valables pour concevoir notre algorithme.

2. Chaque instructeur a effectué au moins une correction de leurs recommandations suite à la révision des réponses détaillées des étudiants, certains en ayant modifié jusqu'à sept, suggérant ainsi que l'accès aux résultats n'est peut-être pas suffisant pour établir l'ensemble des recommandations. De plus, les réponses des instructeurs au questionnaire attestent qu'ils accordent beaucoup d'importance aux réponses détaillées et qu'il sont souvent incertains de leur avis s'ils n'ont pas accès aux réponses détaillées. Ces résultats suggèrent qu'un guide d'étude informatisé risque de ne pouvoir effectuer des recommandations justes sans un diagnostic plus précis basé sur les réponses détaillées

Analyse qualitative des réponses aux questionnaire et entrevues :

3. Tous les instructeurs ont émis des opinions très positives quant à l'utilité anticipée du Guide d'étude Poly pour améliorer l'apprentissage des étudiants au cours de programmation et pour fournir un soutien à l'enseignement.

4. Les réponses aux entrevues ont révélé que l'expérience préalable des instructeurs, lorsqu'ils ont eux-mêmes fourni un plan d'étude, était basé sur une évaluation personnalisée et que le plan s'est avérée effectif selon leur propre appréciation. Nous en concluons qu'un guide d'étude informatisé qui réplique un tel plan devrait aussi être en mesure d'offrir un résultat effectif.

5. Les éléments perçus comme les plus importants d'un guide informatisé sont d'obtenir une évaluation précise de la connaissance et des faiblesses diagnostiquées, de fournir un plan qui optimisera le temps d'étude de l'étudiant et les qualités ergonomiques de l'application.

6. Parmi les avis les plus fréquemment mentionnés par les instructeurs quant au système proposé, on retrouve: la nécessité de fournir un nombre suffisant d'exercices et de fournir une aide aux étudiants pour améliorer leur compétences de résolution de problèmes.

Ces résultats expérimentaux, et plus particulièrement les recommandations fournies par les huit instructeurs, nous serviront de base pour concevoir l'algorithme de recommandation du Guide d'étude Poly.

Algorithme de recommandation et validation

Nous avons conçu un algorithme de recommandation basé sur une Q-matrice. Cet algorithme peut être d'emblée combiné à un engin d'inférence des réponses aux questions comme celui de POKS, et intégré à un guide d'étude. Il vise à émuler les recommandations faites par les instructeurs lors de l'expérience précédente.

L'algorithme consiste à accorder un score, suite à des résultats au test, pour chaque concept auxquelles des questions sont associées. Si la probabilité qu'une question est au-dessus de 0,5, la question est considérée comme réussie. Autrement, elle échouée. Le concept est considéré réussi si la moyenne des questions associées est au delà de 0,5. La matrice est construite manuellement et calibrée manuellement en fonction des résultats d'une première expérience avec 6 étudiants.

Ce modèle est calibré avec les résultats de la première expérience pour les six étudiants, et ensuite validé avec des résultats indépendants de 51 étudiants. En plus de valider l'exactitude, l'expérience consiste à évaluer l'effet du nombre de questions sur la précision des recommandations. Les résultats sont donc analysés selon le nombre de questions administrées. L'évaluation se fait donc à partir de 0 questions admininistrée, où l'évaluation des résultats au test est basé sur la moyenne de l'échantillon et où tous les sujets ont la même recommandation non personnalisée, jusqu'à ce que toutes les questions soient administrée, où le score complet du test est utilisé et où l'on obtient évidemment les résultats les meilleurs. Les questions n'ayant pas été encore été administrées sont déterminées réussies ou non selon leur taux de réussite moyen dans l'échantillon.

Résultats de la seconde expérience

Comme mentionné, une calibration du modèle est effectuée avec les six étudiants et une simulation est faite avec les données d'un corpus de 51 étudiants à titre de validation croisée. Tout comme pour la première expérience, nous mesurons l'exactitude des recommandations en les comparant à celles faites par un instructeur pour trois catégories : « maîtrisé » *(mastered),* « réviser » (*review*) et « étudier » (*focus*).

Les résultats de la simulation démontrent que l'exactitude des recommandations augmente avec le nombre de questions, comme prévu. Lorsque qu'aucune question n'est encore administrée (situation non personnalisée), le taux de succès moyen se situe aux environs de 45%. Il passe à environ 90% lorsque toutes les questions sont posées. Compte tenu que l'accord inter-instructeurs est du même ordre que le résultat de l'algorithme avec toutes les questions, nous concluons que l'algorithme reproduit relativement bien les recommandations des instructeurs.

Les résultats de l'expérience permettent aussi de déterminer l'exactitude attendue des recommandations selon la précision de la prédiction des items d'un engin d'inférence. C'est la situation où seul un sous ensemble du test est administré et l'on tente de faire une recommandation basée sur une inférence quant aux résultats complets. Les résultats démontrent que, par exemple, nous pouvons établir que si un engin d'inférence obtient un score de 90% du résultat aux questions bien prédit, le nombre de recommandations correctes sera de 85 %.

Travaux futurs

Notre étude porte sur l'étude des exigences d'un module de recommandation pour un guide d'étude intelligent et la conception d'un algorithme qui reproduirait aussi fidèlement que possible les recommandations d'experts. Nous avons conçu un algorithme simple qui reproduit relativement fidèlement les résultats expérimentaux de recommandations faites par des experts. Ces travaux peuvent servir de base à la conception du Guide d'étude Poly qui constitue la suite de ce mémoire. Ils permettent notamment de prévoir le taux de recommandations correctes en fonction de la précision de l'engin d'inférence de la connaissance.

Les résultats ont aussi démontré que malgré que les instructeurs jugent qu'il est difficile de porter un diagnostic précis des connaissances acquises sans voir les réponses détaillées, et ainsi de fournir des recommandations valables, il semble que ce problème ne soit pas aussi important que le suggère les résultats initialement. En effet, les résultats de la simulation avec l'algorithme de recommandation demeurent relativement près de ceux des experts. Néanmoins, il est possible que, pour certains sujets ou certaines questions, la nécessité d'avoir un accès à l'analyse du détail des réponses s'avère juste et d'autres études devraient être envisagées pour élucider cette question.

Une des avenues de recherche intéressante à poursuivre est celle d'utiliser des techniques statistiques de forage de données (*data mining*) et d'estimation de paramètres pour créer automatiquement la Q-matrice. Cet objectif est ambitieux mais, déjà, certains chercheurs l'ont abordé avec une technique nommée "Fault Tolerant Teaching (FTT)"(Barnes, 2003).

À la suggestion des instructeurs, il serait utile à un guide d'étude d'avoir un module de génération automatique d'exercices qui permettrait d'avoir un nombre potentiellement illimité d'exercices. Ce facteur est considéré très importants aux yeux des instructeurs. Finalement, et toujours selon l'avis des instructeurs, il faudra porter une attention toute particulière aux facteurs ergonomiques et organisationnels qui entourent le développement et l'implantation du Guide d'étude Poly car même avec un algorithme de recommandation précis et un engin d'évaluation des connaissances efficace, ces facteurs seront aussi déterminants pour le succès d'une telle application.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 General Motivation

Thanks to the rapid growth and popularity of computer networks, especially the Internet and the World Wide Web, Web-Based Educational (WBE) systems are gaining ground over traditional paper-based textbooks. The benefits of web-based education are clear: personalized instructions and freedom of schedule and location for the learner, and economical gains for the institution, such as reduced classroom space, scalability to large groups, and delivery platform independence. Web-based education and training systems installed and supported in one place can be used by thousands of learners all over the world if they are equipped with any kind of Internet-connected computer. That is, learners and educators are able to perform classroom-like tasks: the Web provides the medium and accommodates the educational environment; the educators design the learning experience by preparing the educational material, by deciding on the pedagogical approach, by outlining the learning objectives of the course, by stating how these are fulfilled, and by supporting the learners; lastly, learners are mainly responsible for planning, carrying out, and evaluating their own learning. Besides, unlike printed textbooks, Web-based tutoring systems can incorporate rich multi-media and interactive elements, such as audio, video and animation. Web-based learning systems can add hyperlinks to allow students to click on a link on one web page and immediately be transferred to another page or to other relevant sites. However, since many current Web-based tutoring systems are static HTML Web pages, they suffer from two major shortcomings, namely, they lack interactivity and adaptability (Brusilovsky,1999). Most Web courses present the same static learning materials to students with widely differing knowledge levels of a given subject. Therefore, this kind of system is unable to satisfy the heterogeneous needs of many users (Brusilovsky and Maybury,2002).

Since the dominant educational paradigm has been transferred from "didactic instruction" to "constructivism," an increasing number of researchers believe that using the Web as a technological instructional medium can enhance the learning experience by effectively following a learner-centered pedagogical approach (Lin and Hsieh,2001). Moving in this direction, Web-based Adaptive Educational Systems (AES), which aim to increase the functionality of web by making it more personalized for individual learners, have become a hot research area in recent years.

An Adaptive Educational System usually aims at adapting both usability and learning. Adaptation, in this context, is defined as the concept of making adjustments in an educational environment in order to:

- Accommodate a diversity of learner needs and abilities,
- Maintain the appropriate context for interaction, and
- Increase the functionality of hypermedia by making them more personalized (Brusilovsky,1996; Brusilovsky,1999; Brusilovsky,2001).

Thus, the unique need of learners and the educational implications are very important and should constantly be considered throughout all of the design and development stages of the system.

Many researchers did strive to develop e-learning systems with personalized learning mechanism that can adapt learner's styles, behaviors, interests or habits. However, many of these systems cannot provide learners with effective pedagogical guidance because they usually neglect to validate if the difficulty of instructional materials can match learner's knowledge level. Generally, recommending inappropriate course materials might result in learner's cognitive overhead or disorientation during a learning process. Therefore, more effort should be devoted to personalize knowledge recommendation in the design of AESs.

We briefly outline some of these efforts in section 1.2 and introduce the objectives of this study in section 1.3. Chapter 2 provides a more thorough review of the relevant previous work.

1.2 Preliminary investigation

The focus of our present research is to explore assessment-driven recommendation techniques in an intelligent study guide. Our first consideration was to review existing AES systems and to examine their adaptive and intelligent technologies (Brusilovsky,1999). Among the many Web-based AESs analyzed with respect to the adaptive dimension, technologies from the following have been adopted: Intelligent Tutoring Systems, such as *curriculum sequencing* (DCG, ELM-ART, InterBook, AST, ACE), and *problem-solving support* (ELM-ART, ELM-PE, Lisp-Tutor, SMILE, PROUST, CAMUS II). Also covered were technologies from the following: Adaptive Hypermedia Systems, such as adaptive presentation (MetaDoc, Hypadapter, Anatom-Tutor, C-book, KN-AHS, PUSH, Aha) and *adaptive navigation support* (ISIS-Tutor, Interbook, Hypadapter, ELM-ART, AST, ACE, KBS Hyperbook, AHA).

In curriculum sequencing, the system provides learners with the most suitable and individually planned sequence of knowledge units to learn and with the learning tasks to work with. In *problem solving support*, the main idea is to help learners in solving an educational problem. In adaptive presentation, the content of a hypermedia page is adapted to the learner. Besides, in *adaptive navigation support*; the system alters a number of visible links to support hyperspace navigation. The two most popular technologies, curriculum sequencing and problem solving support, will be reviewed in more detail in section 2.3.

All the technologies stated above compose an integral part of web-based adaptive educational system. Among them, we can take advantage of curriculum sequencing technology to recommend a learning sequence for individual student.

1.3 Objectives

Our work is part of the research for a much bigger project: to develop a study guide for the students in Polytechnique de Montreal, which is also referred as "Poly Study Guide" (Guide d'etude Poly).

This Poly Study Guide is a web-based Adaptive Educational System (AES), and can guide student knowledge remediation by making personalized assessment-driven recommendation. A knowledge recommendation module in this study guide can be guided by an inference engine, such as Partial Order Knowledge Structures (POKS)(Desmarais, Meshkinfam et al.,2006).

This thesis study sets up two objectives:

Objective 1: To investigate the requirement for the recommendation module in our intelligent study guide,

In order to investigate the requirement for the recommendation module, we realize an experiment with eight experienced instructors, explore the process of real one-on-one tutoring and collect information about professional knowledge diagnosis and study plan recommendations for each individual student. Both quantitative and qualitative data derived from this experiment are analyzed.

Objective 2: To devise an appropriate recommendation algorithm that can grant the study guide the ability to diagnose knowledge states and make individual study plan, and to investigate the effectiveness of this knowledge recommendation algorithm in our proposed intelligent study guide.

We emulate the observed recommendation results with instructors (See *Objective 1*) by devising a q-matrix and a corresponding algorithm in the recommendation module that can be integrated with an adaptive assessment module, such as POKS, to make personalized assessment-driven recommendation. Finally, we implement a simulation test to validate the effectiveness of the recommendations from our intelligent study guide.

1.4 Organization of the thesis

Chapter 2 provides background information for intelligent educational systems and investigates the intelligent technologies applied in those systems.

Chapter 3 further reviews existing studies about study recommendation techniques in study guides that are related to our work.

Chapter 4 takes the example of Partial Order Knowledge Structures (POKS) framework to investigate the techniques that can offer an adaptive assessment engine the ability to make assessment-driven knowledge recommendation.

Chapter 5 presents an experimental study with experienced instructors in order to investigate the process of real one-on-one tutoring and collecting information about professional knowledge diagnosis and study plan recommendations

Chapter 6 presents the design and implementation of a recommendation module derived from the data in the experiment with instructors. Besides, a simulation test was conducted to examine the effectiveness of this module in our proposed intelligent study guide.

Chapter 7 concludes the thesis and discusses future work.

CHAPTER 2
REVIEW OF INTELLIGENT EDUCATIONAL SYSTEM

2.1 Introduction

This chapter provides the background information for intelligent educational systems. In section 2.2, we present the background knowledge for Intelligent Tutoring Systems (ITS), an earlier kind of Web-based Adaptive Educational System (AES). Firstly, we review the history of Intelligent Tutoring Systems and discuss their general framework and the function for each component. In addition, we focus on the student model that is the key aspect of Intelligent Tutoring Systems needed to realize one-on-one tutoring. In section 2.3, we review the technologies applied in Web-based AES, especially, the two classical approaches: problem solving support and curriculum sequencing. Finally, we conclude this chapter in section 2.4.

2.2 intelligent tutoring systems

2.2.1 History of intelligent tutoring systems

In this section, we review the history of Intelligent Tutoring System, an original form of web-based Adaptive Educational System (AES), and discuss their development in the context of Artificial Intelligence (AI) and educational theory.

Intelligent Tutoring Systems have an interesting history, originating in the Artificial

Intelligence (AI) movement of the late 1950's and early 1960's (Urban-Lurain,2004). Researchers such as Alan Turing, Marvin Minsky, John McCarthy and Allen Newell thought that computers that could "think" as humans do were just around the corner.

Since the 1960s, researchers have created a number of Computer Assisted Instructional (CAI) systems (Uhr,1969; Sleeman and Brown,1982; Urban-Lurain,2004). CAI systems generated sets of problems designed to enhance student performance in skill-based domains, primarily arithmetic and vocabulary recall (Urban-Lurain,2004). Martin & VanLehn (1995) claim that CAI systems present instructional materials in a rigid tree structure to guide the students from one content page to another depending on their answers, as illustrated in Fig. 1. Such systems, grounded in didactic instruction, do not generate flexible instructional plans. Instead, they follow a pre-specified and fixed plan. Moreover, CAI systems are not adaptive and unable to dynamically provide the same kind of individualized attention that students would receive from human teachers (Bennett,1999).

Throughout 1970's and into the 80's, with the development of Piaget's theories of constructivism and Chomsky's Information Processing (IP) theory, AI researchers made a lot of attempts to reduce the drawbacks of CAI systems stated above. In 1982, Sleeman and Brown reviewed the state of the art in computer-aided instruction and first coined the term *Intelligent Tutoring Systems* (ITS) to describe some evolving systems and distinguish them from the previous CAI systems. Intelligent Tutoring Systems present educational materials in a flexible and personalized way that is similar to one-on-one tutoring (Brusilovsky,1999). In particular, ITSs have the ability to provide learners with tailored instructions and feedback.

One of the key elements that distinguishes ITSs from more traditional CAI systems is ITSs' capability to dynamically maintain a model of a student's reasoning and learning that keeps track of a student's knowledge during the study (Shute and Psotka,1996). As noted by Shute and Psotks(Shute and Psotka,1996), ITSs must be able to achieve three main tasks:

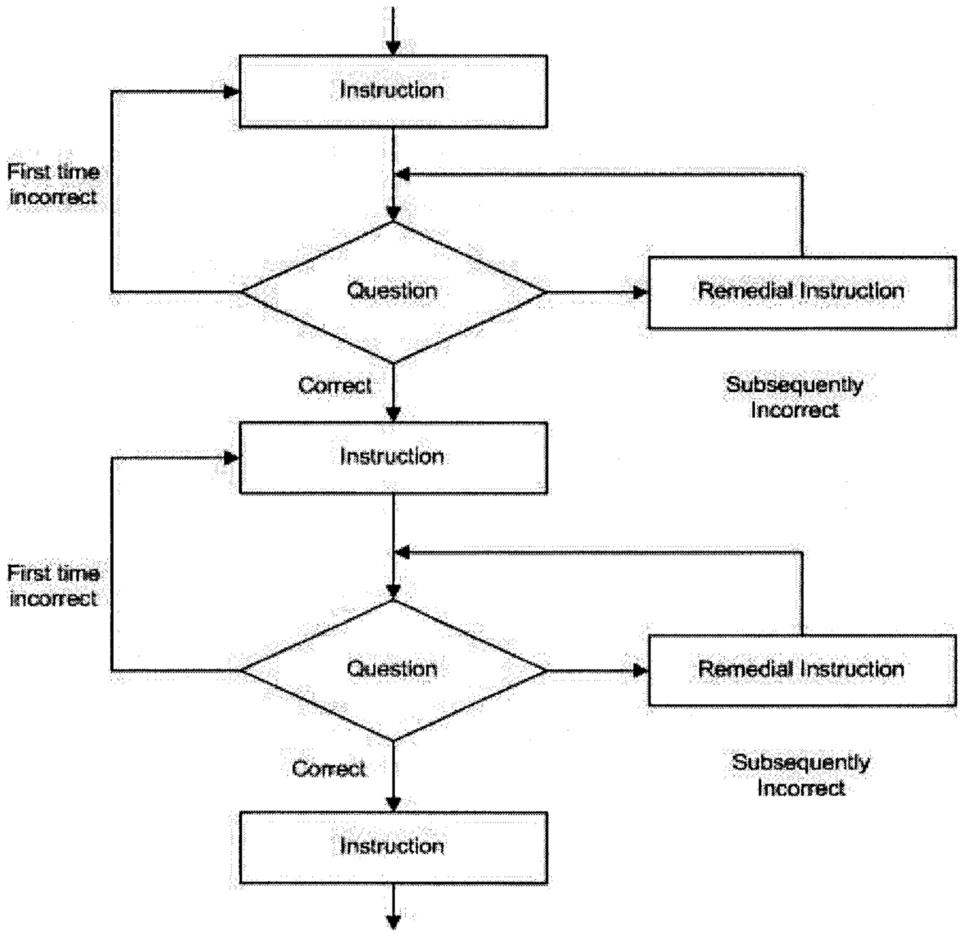


Figure 1 Tree structure for one traditional CAI system to guide the student in navigating through the instructional materials[1].

[1] http://www.cs.mdx.ac.uk/staffpages/serengul/Traditional.Computer.Aided.Learning.Systems.htm April 28, 2004.

(i) Accurately diagnose a student's knowledge level using principles rather than preprogrammed responses;

(ii) Decide what to do next and adapt instruction accordingly;

(iii) Provide feedback.

This kind of diagnosis and adaptation implements the individual one-on-one tutoring that is demonstrated as the most effective teaching method (Bloom,1984). ITSs have been shown to be highly effective in increasing students' performance and motivation levels compared with traditional instructional methods (Shute and Glaser,1990; Koedinger, Anderson and Hadley,1997).

2.2.2 Key components of ITS

ITS uses the knowledge about the domain, the student, and about teaching strategies to support flexible individualized learning and tutoring. Researchers typically separate an ITS into several different parts, and each part plays an individual function. Usually, most ITSs have four common major components (Sleeman and Brown,1982): (1) knowledge domain; (2) student model; (3) teaching Strategies; and (4) user interface, as illustrated in Fig.2:

Main ITS Components

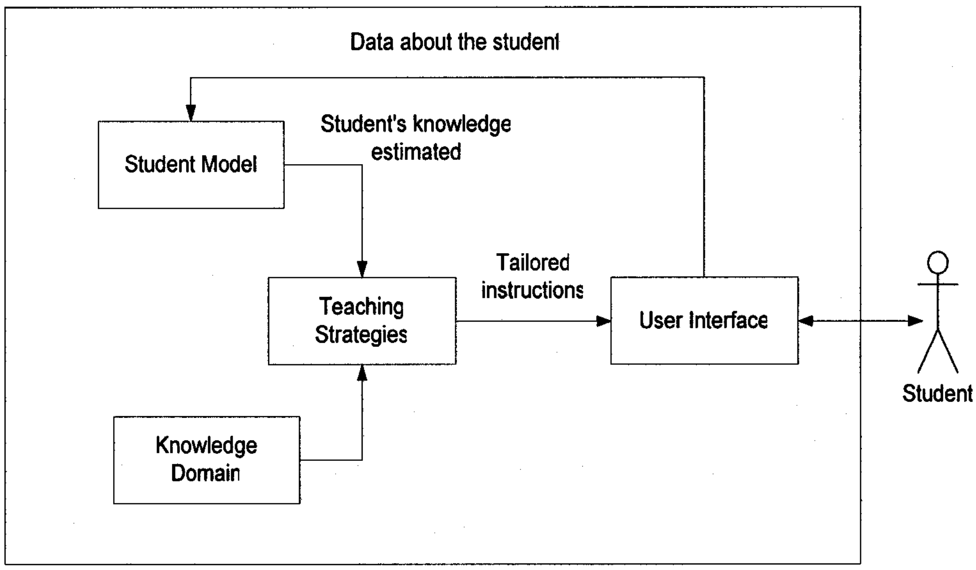


Figure 2 The major components of most Intelligent Tutoring Systems

2.2.2.1 Knowledge domain

The knowledge domain stores educational materials that the students are required to study for the topic or curriculum being taught.

2.2.2.2 Student model

Student modeling remains at the core of ITS research (Holt, Dubs, Jones and Greer,1994). The student model stores information that is specific to each individual learner. Without an explicit student model, the teaching strategies component is unable to make decisions to adapt instructional content and guidance (see Fig.2) and is forced to treat all students similarly.

Mitchell argues that an ITS must model the world, the learner, and the teacher-learner interaction (Mitchell and Grogonon,1993). According to Wenger (Wenger,1987), student models have three tasks.

(1) They must gather data from and about the learner. This data can be explicit -- asking the student to solve specific problems -- or implicit -- tracking the students navigation and other interactions and comparing them to information about similar learner responses.

(2) They must use that data to create a representation of the student's knowledge and learning process. This often takes the form of "buggy" models that represent the student's knowledge in terms of deviations from an expert's knowledge. The system then uses this model to predict what type of response the student will make in subsequent situations, compares that prediction to the students' actual response, and uses that information to refine the model of the student.

(3) The student model must account for the data by performing some type of diagnosis, both of the state of the student's knowledge and in terms of selecting optimal pedagogical strategies for presenting subsequent domain information to the student. One of the biggest challenges is to account for "noisy" data, the fact that students do not always respond consistently, particularly when their knowledge is fragile and they are uncertain about the correct responses.

From the three tasks, we can see that modeling the student within an intelligent tutoring system involves a good deal of inherent uncertainty (Conati, Gertner and Vanlehn,2002). Thus, one of the biggest challenges in designing ITSs is the effective assessment and representation of the student's knowledge state and specific needs in the problem domain based on uncertainty information. Over the past decade the question of how to manage uncertainty has been a rapidly expanding and increasingly mainstream research topic in Artificial Intelligence. Various approaches in Artificial Intelligence have been proposed for uncertainty reasoning (Nilson,1998), including rule-based systems (Buchanan and Sortliffe,1985), fuzzy logic (Klir and

Yuan,1995), DempsterShafer theory of evidence, and neural networks. Bayesian networks (Pearl,1988), in particular, are a powerful approach for uncertainty management in Artificial Intelligence (Wong, Butz and Wu,2000; Wong and Butz,2001).

2.2.2.3 Teaching strategies

The teaching strategies component refers to educational techniques for teaching. For instance, the component decides when to present a new topic, how to provide recommendations and guidance, and which topic to present. As mentioned earlier, the assessment result of the student model is input to this component, so the system's pedagogical decisions reflect different needs of students. Thus, this component needs to take appropriate actions to manage one-on-one tutoring, such as switching teaching strategies and using a variety of teaching approaches at the appropriate times according to the student's particular needs and problems.

2.2.2.4 User interface component

The user interface allows communication between student and the other modules of an ITS. The dialogue and the screen layouts are controlled by this component.

Research from the human factors and software design disciplines is applicable, but the pedagogical implications of an ITS interface must also be considered. Definitely, a well-designed interface can enhance the capabilities of an ITS by allowing the system to present instructions and feedback to the student in a clear and direct way.

2.3 Intelligent technologies for Web-based adaptive educational systems

Thanks to high accessibility and great flexibility of Internet, web-based Adaptive Educational System (AES) inheriting from Intelligent Tutoring System (ITS) and Adaptive Hypermedia System (AHS) became a hot research and development area. In this section, we present two of the most popular intelligent technologies applied in Web-based AES.

2.3.1 Problem solving support

For many years, problem solving support was considered as the primary duty of an ITS and a main value of an ITS technology (Brusilovsky,1999). This technique is also widely used among a lot of web-based adaptive educational systems. The objective of problem solving support is mainly to provide students with intelligent help for each step when solving an educational problem. For example, when a student encounters difficulties on one step, this technology can provide a hint showing the next correct solution step for the student, or offering appropriate error feedback. The major challenge for this technique is to interpret the student's actions and infer the solution plan that the student is currently following based on a partial sequence of observable actions. That is, the system needs to understand the student's plan, and apply this understanding to provide help. Examples of this type of ITS are (Anderson, Conrad and Corbett,1989; Gertner, Conati and Vanlehn,1998; Schulze, Shelby, Treacy, Wintersgill, Vanlehn and Gertner,2000; Johnson,2001; Liu, Zheng, Ji, Yang and Yang,2001; Sykes and Franek,2003).

However, the purpose of our proposed web-based intelligent study guide is quite different from providing problem-solving support; it is to diagnose students' knowledge states and provide them with suitable individual study plan. The technique that we are exploring in our research is akin to curriculum sequencing, which is presented in next section.

2.3.2 Curriculum sequencing

Curriculum sequencing is now the most popular and important technology in web-based adaptive educational systems (Brusilovsky,1999). The goal of the curriculum sequencing technology (also referred to as instructional planning technology) is to provide the student with the most suitable individually planned sequence of knowledge units to learn and sequence some learning tasks (examples, questions, problems, etc.) to work with (Brusilovsky,1999). Some examples of curriculum sequencing are (Barr, Beard and Atkinson,1976) and (Brusilovsky,1999).

There are two essentially different kinds of sequencing: active and passive (Brusilovsky,1999). Active sequencing indicates a learning goal (a subset of domain concepts or topics to be mastered) for an individual and it can create the best individual path to achieve the goal. Passive sequencing is a reactive technology and does not require an active learning goal. It starts when the user is not able to solve a problem or answer a question (questions) correctly. Its goal is to offer the user a subset of available learning material, which can fill the gap in student's knowledge of resolving a misconception. Active sequencing is the dominant type of sequencing. It is provided by numerous systems such as ELM-ART-II, AST, ADI, ART-Web, ACE, KBS-Hyperbook, and ILESA. Only a few systems (InterBook, PAT-InterBook, CALAT, VC Prolog Tutor, and Remedial Multimedia System) can perform passive remedial sequencing. Active sequencing technique is a major concern of our proposed POKS intelligent study guide. We attempt to design an adaptive educational system that can recommend appropriate learning sequence by generating two lists, "review" and "focus", from learning materials in order to guide students to the learning goal- the whole set of domain concepts.

In the context of Web-based education, recommending appropriate learning sequencing becomes very important to guide a student through the hyperspace of available information. Web-based learning students usually work alone without a teacher's instructional assistance and they study the subject at their own pace. As a result, appropriate learning sequencing recommendations are essential in order to enable each student to learn the subject in the most beneficial and individualized way (Brusilovsky,1999). It's not surprising that, not only is recommending sequence the oldest, but also the most popular of Web-based AES. We will further review existing studies about knowledge recommendation techniques in study guides that are related to our work in Chapter 3.

2.4 Summary

This chapter reviews the background information for our research.

Intelligent Tutoring System (ITS), as an earlier form of Web-based Adaptive Educational System (AES), derived from Computer Assisted Instruction (CAI) system. ITSs have four common major components (Sleeman and Brown,1982): knowledge domain; student model; teaching strategies; and user interface, among which student modeling is always at the core of ITS research. Two most popular intelligent technologies applied in web-based AES are problem solving support and curriculum sequencing. How to realize curriculum sequencing in the recommendation module of a study guide is our major concern. In next chapter, we will further review existing studies about knowledge recommendation techniques in study guides that are related to our work

CHAPTER 3

STUDY RECOMMENDATION

3.1 Introduction

STUDY recommendation is an essential component of an intelligent study guide. (Khuwaja, Desmarais and Cheng,1996) defined that the goal of an intelligent guide is to provide pedagogical guidance based on user knowledge assessment, such as recommending appropriate educational materials for a number of domains that require the user to master a number of concepts or skills to achieve a satisfactory level of competence in the domain. In this chapter, we further review existing studies about knowledge recommendation techniques in intelligent study guides that are related to our work.

In section 3.2, we clarify the differences between the knowledge recommendation techniques in our research and those applied in collaborative filtering systems. In section 3.3, we analyze two major inference frameworks for assessment-driven knowledge recommendation: Bayesian Networks and Knowledge Space Theory. More specifically, we explain the two techniques by illustrating two corresponding existing intelligent study guides, ALEKS and BITS, which are most recent and closely related to our work. An emphasis is placed on their underlying theoretical grounding and inference engines. Section 3.4 concludes this chapter.

3.2 Study recommendation techniques

As we stated in chapter 1, our proposed POKS Intelligent Study Guide can guide student knowledge remediation by making personalized assessment-driven recommendation. The assessment-driven recommendation defined here for our study guide in this research is different from the recommendation provided by those collaborative filtering systems (also referred to as recommendation systems).

Recommendation systems are typically based on human-machine interaction mediated by intelligent agents, or other decentralized components (Jonhson, Rasmussen, Joslyn, Rocha, Smith and Kantor,1998) and come in several varieties: content-based recommendation; collaborative recommendation; structural recommendation; collective recommendation. Some intelligent learning systems take advantage of one or a few recommendation techniques stated above to provide adaptive knowledge recommendation to learners. For example, Tang and Mccalla (2004) propose an evolving web-based learning system that can adapt itself not only to its users, but also to the open Web. More specifically, the novelty with respect to the system lies in its ability to find relevant content on the web, and its ability to make smart, adaptive recommendations based on the system's observations of learners' activities throughout their learning and the accumulated ratings given by the learners. Two of the major techniques that are adopted in this evolving web-based learning system include collaborative filtering and data clustering. Besides, Zaiane (2003) suggests the use of structural recommendation and web mining techniques to build an intelligent agent that could recommend on-line learning activities or shortcuts in a course web site based on learners' access history to improve course material navigation as well as assist the online learning process.

However, the collaborative filtering techniques are out of our research scope currently. We place an emphasis on the approaches that can make personalized recommendation for learning goals and materials based on knowledge assessment.

Without knowledge assessment, there is no way of measuring the result of teaching, or tailoring further education. The Alberta Research Council (1995) reports " in order to allow instruction to be individually designed, it is first necessary to capture the student's understanding of subject."(Stauffer,1996)

Personalized recommendation, based on a good assessment that does not just report those questions a student missed, but offers a stronger reflection of the skills and understanding underlying a student's performance, can undoubtedly promote learning efficiency and effectiveness (Dietel, Herman and Knuth,1991).

Generally, the adaptive recommendation mechanisms applied in many existing intelligent study guides are guided by fine-grained knowledge assessment modules based on a variety of theoretical frameworks for uncertainty management such as Bayesian Networks(Butz, Hua and Maguire,2006), Knowledge Space Theory (Falmagne, Doignon, Cosyn and Thiery), or Fuzzy Item Response Theory (FIRT) (Chen, Duh and Liu,2004), etc.

In next section, we will introduce some study guides to illustrate two major assessment-driven recommendation techniques based on the theories of Bayesian Networks (BN) and Knowledge Spaces Theory.

3.3 Techniques for Assessment-driven knowledge recommendation

In this section, we illustrate the theoretical frameworks of Bayesian Networks and Knowledge Spaces by mainly analyzing the inference engines and the process of making assessment-driven recommendation of two specific web-based intelligent study guides: Bayesian Intelligent Tutoring System (BITS) and Assessment and LEarning in Knowledge Spaces (ALEKS) system.

3.3.1 Bayesian Networks

Researchers have applied Bayesian networks to many tasks, including student monitoring(Johnson,2001; Liu, Zheng et al.,2001), e-commerce (Liu, Zheng et al.,2001; Robles, Lfarranaga, Menasalvas, Perez and Herves,2003), and multi-agents (Lee, Sung and Cho,2001; Wang and Vassileva,2003). Villano (1992) first suggested applying Bayesian networks in Intelligent Tutoring Systems. The assessment system proposed by Martin and Vanlehn (1995) is focused solely on assessing what a student knows. Zapata and Greer (2001) designed a system to visualize Bayesian student models. One advantage of graphical representations of student models is to help instructors determine the learning deficiencies for a student. There have been several other recent efforts to apply Bayesian networks to student modeling in Web-based tutoring systems (Gertner, Conati et al.,1998; Henze and Nejdl,2001; Johnson,2001).

A Web-based intelligent tutoring system (BITS) (Butz, Hua et al.,2006) is a recent study guide that employs Bayesian Networks as an inference engine to guide the students' learning processes. BITS is very similar to our proposed system because it not only assesses what a student knows, but, in addition, it recommends an appropriate learning sequence and assists the student in navigating the unknown concepts. Thus, in next section, we describe BITS in more details.

3.3.1.1 BITS

BITS is an intelligent study guide for computer programming. It can assist a student in navigation through the online materials and can recommend learning goals, and generate appropriate reading sequences.

The decision making process conducted in BITS takes full advantage of Bayesian networks, which is a formal framework for uncertainty management in Artificial Intelligence based on the Bayesian framwork. More specifically, Bayesian networks can help BITS meet two major objectives: modeling the structure of a problem domain, and tracking student knowledge regarding each concept in the problem domain.

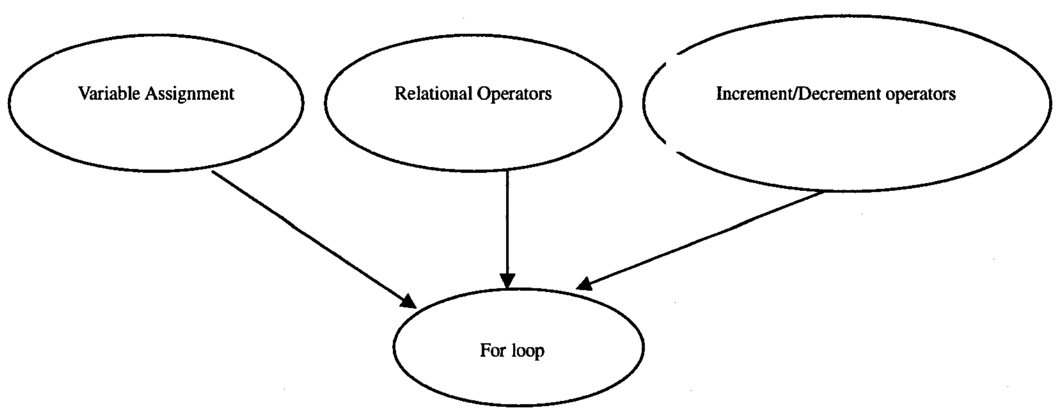


Figure 3 Modeling the prerequisite concepts of the “For Loop” construct

Firstly, using a Bayesian Network, the prerequisite relationships among concepts identified for a computer science introductory course can be represented directly and clearly (See Figure 3). A directed edge from one concept (node) to another can be added if knowledge of the former is a prerequisite for understanding the latter.

Furthermore, BITS can track a student's knowledge by obtaining relevant evidence to update the Bayesian network. There are two methods of obtaining new evidence: a student's direct reply to BITS query if this student knows a particular concept; a sample quiz result for the corresponding concept to determine whether or not a student has understood a particular concept.

Using the state of the Bayesian network regarding the knowledge of the student, BITS can offer tailored pedagogical options that support the individual student. There are three kinds of adaptive guidance that BITS can provide: navigation support, prerequisite recommendations for problem solving, and the generation of a learning sequence when studying a particular concept.

BITS can address the problem of Web-based learners' unproductive navigation, and refocus them on their study objectives by making the study guide adaptable to different types of learners (Butz, Hua et al.,2006). However, the inference engine in BITS is very complex to build: Manually constructing the DAG (Directed Acyclic Graph) for a Bayesian network that contains many nodes is a major knowledge engineering effort and it is very domain specific. In addition, the disadvantage of the assessment offered by BITS is that it is difficult to measure whether a student really understands the knowledge by his/her reply to a query or just a sample quiz. Thus, it is imperative to explore an easy-to-build and fine-grained assessment module for our proposed intelligent study guide.

3.3.2 Knowledge Space Theory

Some intelligent tutoring systems are based upon theoretical work in Cognitive Psychology and Applied Mathematics in a field of study called "Knowledge Space Theory" (Falmagne, Doignon et al.), such as ALEKS[2] (Assessment and LEarning in Knowledge Spaces) system and RATH[3] (Relational Adaptive Tutoring Hypertext WWW-Environment). In this section, we take ALEKS as a most recent and related example to illustrate how to apply knowledge space theory to guide learner's study by making assessment-driven recommendation.

3.3.2.1 ALEKS

ALEKS (Assessment and LEarning in Knowledge Spaces) system is currently, in our view, the most powerful commercial intelligent study guide on the world-wide-web. By knowing exactly which math concepts the student has mastered, which are shaky, and which are new but within reach, ALEKS is able to intelligently recommend those concepts that the student is most ready to learn.

ALEKS' assessment module assesses the student's current knowledge of the subject by asking the student a small number of questions (approximately 15-25 for Arithmetic). ALEKS chooses each question on the basis of the individual student's answers to all the previous questions. Each student's set of assessment questions is unique. After it quickly and efficiently develops a detailed and comprehensive map of the student's mastery of the subject, ALEKS is equipped with a precise picture of the student's knowledge state, and the student is offered a choice among the concepts the student is most ready to learn.

---

[2] http://www.aleks.com

[3] http://wundt.kfunigraz.ac.at/rath

The concept of 'knowledge state' is at the basis of the Knowledge Spaces theory. The set of possible knowledge states is defined as the complete set of problems that an individual is capable of solving in a particular topic, such as Arithmetic or Elementary Algebra. That set of possible knowledge states defines the knowledge space for a given domain. The inference engine in ALEKS must uncover the particular state of the student being assessed, among all the feasible states. The result of an assessment consists in two short lists of problems that may be labeled: 'what the student can do' and 'what the student is ready to learn'. These two lists specify the exact knowledge state of the individual being assessed. Besides, the list 'what the student is ready to learn' is exactly the recommendation given by the system.

To build a knowledge space for ALEKS, experts, such as seasoned teachers or textbook writers, were enrolled to find the possible knowledge states for each domain, defined as a *knowledge structure*. A first draft of a knowledge structure can be defined by domain experts and must be refined by a painstaking analysis of student data.

Suppose that a satisfactory knowledge structure has been obtained. The task of the assessment is to uncover, by efficient questioning, the knowledge state of a particular student under examination. The procedure of uncovering a knowledge state all pertains to the scheme outlined in Fig 4 (Falmagne, Doignon et al.):

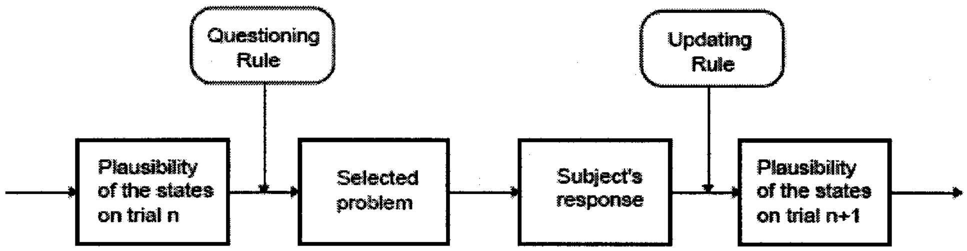


Figure 4 Diagram of the transitions in an assessment procedure

At the beginning, certain likelihood is assigned to each of the knowledge state and will be updated accordingly when new evidence comes. As to the questioning rule, the problem p_n is chosen so as to be 'maximally informative.' This is interpreted to mean that, on the basis of the current likelihoods of the states, the student has about a 50% chance of knowing how to solve p_n. In other words, the sum of the likelihoods of all the states containing p_n is as close to 0.5 as possible. If several problem types are equally informative (as may happen at the beginning of an assessment), one of them is chosen at random. The student is then asked to solve an instance of that problem, also picked randomly. The student's answer is then checked by the system and the likelihood of all the states are modified according to the following *updating rule*. If the student gave a correct answer to p_n, the likelihoods of all the states containing p_n are increased and, correspondingly, the likelihoods of all the states not containing p_n are decreased (so that the overall likelihood, summed over all the states, remains equal to 1). The probability update algorithm of ALEKS is unfortunately not described in the paper (Falmagne, Doignon et al.). A false response given by the student has the opposite effect: the likelihoods of all the states not containing p_n are increased, and that of the remaining states decreased. The assessment procedure stops when two criteria are fulfilled: (1) the entropy of the likelihood distribution, which measures the uncertainty of the assessment system regarding the student's state, reaches a critical low level, and (2) there is no longer any useful question to be asked (all the problems have either a very high or a very low probability of being responded to correctly). At that moment, a few likely states remain and the system selects the most likely one among them.

The success stories and quantitative data reported by the ALEKS research group show that ALEKS is an effective, proven system that addresses many of the concerns in educational domains. However, the underlying Knowledge Space Theory requires domain specific modeling and involves intensive knowledge engineering effort, much like Bayesian Networks. Hence, some researchers have endeavored to make this knowledge engineering process less tedious and semi-automatic. For example, a QUERY procedure was designed to systematically question an expert, and construct a unique knowledge space consistent with the expert's responses (Kambouri, Koppen, Villano and Falmagne,1994). However, it remains a knowledge and time intensive process. In our research, we attempt to explore a cost-effective intelligent study guide that is not only easy to build but also can deliver precise assessment-driven knowledge recommendation and provide effective pedagogy guidance. The recommendation in our proposed intelligent study guide is based on the assessment from much easier and effective student-modeling frameworks such as Partial Order Knowledge Structure (POKS). In chapter 4, we take POKS as an example of assessment engine to investigate the methods of knowledge recommendation based on an adaptive assessment.

3.4 Summary

Personalized recommendation based on a good assessment can promote learning efficiency and effectiveness. We focus our research on this assessment-driven recommendation techniques, which is different from the recommendation provided by those collaborative filtering systems.

We illustrate two major inference frameworks for assessment-driven knowledge recommendation, Bayesian Networks and Knowledge Space Theory, by explaining the inference engines of two recent intelligent study guides, ALEKS and BITS.

Bayesian Networks and Knowledge Space Theory require domain specific modeling and involve intensive knowledge engineering effort. Thus, the recommendation in our proposed intelligent study guide could be based on the assessment from a much easier and effective student modeling technique, such as Partial Order Knowledge Structure (POKS) or Item Response Theory (IRT). In chapter 4, we take POKS as an example to investigate the techniques that can offer an assessment engine the ability to make knowledge recommendation.

CHAPTER 4

GENERATING RECOMMENDATION FROM A SIMPLE ASSESSMENT ENGINE

4.1 Introduction

In last chapter, we review the assessment-driven recommendation techniques in study guides. We concluded that many student modeling techniques such as Bayesian Networks and Knowledge Space Theory are too expensive to build. Thus, we can take advantage of a much simpler framework such as POKS (Partial Order Knowledge Structures), as the inference engine of our proposed intelligent study guide. In this chapter, we take POKS as an example to investigate the techniques that can offer an assessment engine the ability to make assessment-driven knowledge recommendation.

In section 4.2, we briefly review the underlying theory of POKS model. In section 4.3, how POKS can lead to knowledge recommendation based on more general concept and skill mastery assessment is further discussed. An emphasis is placed on Q-matrix method. Finally, we conclude this chapter in section 4.4.

4.2 Adaptive assessment by POKS

Approaches such as Bayesian networks (BN) are considered highly powerful modeling and inference techniques. However, POKS is a simpler Bayes posterior probability update approach under strong independence assumption that does not require great knowledge engineering effort. In this aspect, it can be considered more advantageous.

The most distinctive characteristic of Partial Order Knowledge Structure (POKS) (Desmarais, Maluf and Liu,1996; Desmarais and Pu,2005a) is that it permits the inference of known or unknown items based on item to item structure. It derives from the work of Falmagne et al. (1990) and Doignon and Falmagne(1999). Others such as Kambouri, Koppen, Villano, and Falmagne (1994) have worked towards using the structural characteristics of item-to-item structures to infer an individual's knowledge state.

Item-to-item relations are grounded in the Theory of Knowledge Spaces (Falmagne, Koppen, Villano, Doignon and Johannesen,1990). However, POKS structure is built with different assumption. That is, knowledge states defined by POKS are closed under union and intersection and ignore the possibility of representing alternate prerequisite knowledge items (Desmarais, Maluf et al.,1996; Desmarais, Fu and Pu,2005). Such structures can be represented by a DAG because of the assumption of closure under intersection and union (Desmarais, Meshkinfam et al.,2006). This assumption allows considerable reduction of the data set size required (Desmarais and Pu,2005b).

POKS abandons the ability to model alternative prerequisites, but it permits the complete automation of the knowledge structure induction process. Determining which items are linked together is based on the POKS network induction algorithm (Desmarais, Maluf et al.,1996). The POKS induction algorithm relies on a pair-wise analysis of item-to-item relationships. After the POKS network structure is constructed, it can be used for knowledge inference. Computation of the nodes' probabilities is essentially based on standard Bayesian posteriors (Desmarais, Meshkinfam et al.,2006).

4.3 From item results to concept and skill prediction

Although some simple assessment engines do not need knowledge engineering effort, they contain no information such as concepts and misconceptions. Thus, such an assessment cannot offer a strong reflection of the skills and understanding underlying a student's performance. Therefore, we need to find an appropriate approach that can offer an assessment engine the ability to diagnose knowledge states and make corresponding recommendations Here we still take POKS as an example to present some approaches that can help to infer mastery of concepts from observable item nodes, since POKS framework builds relations merely among observable question items. A focus is placed on q-matrix method

4.3.1 Q-matrix and Rule Space

In this section, we discuss Tatsuoka (Tatsuoka,1983)'s rule space and q-matrix as a simple and widely accepted method that can make concept and skill prediction.

Tatsuoka (Tatsuoka,1983) introduced the concepts of rule space and q-matrix. The main goal was diagnosis of students' misconceptions, which could be used to guide remediation, assess group performance as a measure of teaching effectiveness, and discover difficult topics (Birenbaum, Kelly and Tatsuoka,1993). The q-matrix is a binary matrix showing the relationship between test items and latent or underlying attributes, or concepts (Birenbaum, Kelly et al.,1993). Students were assigned knowledge states based on their test answers and the constructed q-matrix. An example of a binary q-matrix is given in Table 1. A q-matrix, or "attribute-by-item incidence matrix", contains a one if a question is related to the concepts, and a zero if not. For example, in this q-matrix, question q2 and q6 are both related by concept con1, while q1 is also related to q2 and q4 by concept con2. An alternative to the Q-matrix is to decompose the mastery of a given concept as a weighted mean of items, much in the same manner as every teacher does when points are allotted to different test items in an exam.

Table 1 Example q-matrix

	Questions					
	Q1	Q2	Q3	Q4	Q5	Q6
Con1	0	1	0	0	0	1
Con2	1	1	0	1	0	0
Con3	1	1	1	0	0	0
Con4	1	0	1	0	0	0

Rule space is a statistical methodology for classifying students' responses to a set of items into one prespecified attribute-mastery pattern (Birenbaum, Kelly et al.,1993). In practice, a domain expert and cognitive scientist would identify a q-matrix in a rule space. However, a student's actual mastery or non-mastery of a set of attributes cannot be measured directly, but must be inferred from the student's pattern of response to the items. In an ideal case, a student who had mastered some attributes would answer correctly only those items that contain attributes that he or she had mastered and answer incorrectly those items that contain at least one attribute that he or she had not mastered. Such a student would produce an ideal item-response pattern. Within rule space, specialized functions, called Boolean Description Functions (BDF), are used systematically to determine the knowledge states of interest (i.e., those that describe ideal behavior in terms of attributes) and to map them into ideal item-response patterns (Varadi and Tatsuoka,1989; Tatsuoka,1991). Rule space then plots the ideal item-response patterns in terms of two variables: θ (theta) and ζ (zeta). θ is the ability continuum derived from an item-response (IRT) analysis (Lord and Novick,1968). A student of high ability who gets some easy items incorrect or a student of low ability who gets some hard items correct would be measured high on an " unusualness of response" scale, which is what ζ is (Tatsuoka and Linn,1983; Tatsuoka,1984). ζ is the second dimension that rule space uses to describe students' responses.

Rule space entails a statistical pattern-classification approach. Its accuracy of classification depends on how well the items are written, how well they test (as unambiguously as possible) the attributes that were established by the domain expert, and the amount of error in the student's responses. For areas that are well-defined (e.g., subtraction of fractions), rule space has been shown to perform quite well (Tatsuoka,1990; Tatsuoka and Tatsuoka,1992).

Tatsuoka's Q-matrix and rule space research showed that it is possible to automate the diagnosis of student knowledge states, based solely on student item-response patterns and the relationship between questions and their concepts. Although it also involves a knowledge engineering effort, this Q-matrix approach has the advantage of being readily understood by teachers who frequently go through this process of determining which test items assess which concepts or topics. Therefore, we were inspired to take advantage of Q-matrix method to introduce concept nodes in POKS. However, we planned to devise a new algorithm by emulating a real student-tutoring process, instead of applying rule space statistical methodology. In our research, we administered an experiment with instructors and took their recommendations for each individual student as expert knowledge assessment. Then, we create a Q-matrix and an appropriate algorithm by simulating the experimental results. We describe the details of the experiment with instructors and the simulation test separately in Chapter 5 and Chapter 6.

4.4 Summary

As we stated in chapter3, many student modeling techniques such as Bayesian Networks and Knowledge Space Theory are too expensive to build. Thus, in our research, we attempt to explore a cost-effective intelligent study guide that is not only easy to build but also can deliver precise assessment-driven knowledge recommendation and provide effective pedagogy guidance. In this chapter, we take Partial Order Knowledge Structures (POKS) model as a good example for adaptive assessment since it is based on item-to-item structures that can be learned from small data samples.

POKS makes several strong assumptions to reduce complexity and it allows the Bayesian modeling of item-to-item knowledge structure in accordance to the Theory of Knowledge Spaces (Falmagne, Koppen et al.,1990). The POKS network induction algorithm relies on a pair-wise analysis of item-to-item relationships. Furthermore, the implementation of evidence updating is consistent with the Bayesian posterior probability computation in single layered networks and corresponds to the posterior probability update.

However, the network of POKS is defined solely over the test items and no concept nodes are included. Some methods that can offer POKS the ability to yield more general concept and skill mastery assessment and recommendation are analyzed in chapter 4.3. In our research, we take advantage of the Q-matrix method and create a recommendation module to introduce concept nodes in POKS by emulating experimental results derived from professional instructors. An experiment with instructors and a simulation test will be presented in Chapter 5 and Chapter 6.

CHAPTER 5

EXPERIMENT WITH INSTRUCTORS

5.1 Introduction

In this chapter, we present the methods, procedures and results of an experiment with instructors. The objective of this experiment is to investigate the requirements for the recommendation module of our proposed intelligent study guide. More specifically, we collect information in this experiment about professional knowledge diagnosis and study plan recommendations for each individual student, and examine the agreement among recommendations from different instructors in order to determine what recommendations a study guide should deliver. It sets the desired result of a study guide. Besides, we also attempt to determine the value of detailed answers in terms of improving recommendations and examine the potential effects of the proposed system on students' performance. The experimental results will serve as data for devising a recommendation module in our proposed study guide. More details about emulating experimental results will be presented in next chapter.

In section 5.2, we present a pilot survey study aiming to validate the research design. In section 5.3, we present the design of the experiment. The detailed methods and procedures of this experiment are reported. In section 5.4, we analyze the data derived from this experiment. Both quantitative and qualitative analyses are presented. In section 5.5, a summary concludes the chapter.

5.2 A pilot study

We conducted a pilot study before the principal experiment, aiming to explore the effective procedure for gathering professional knowledge recommendation from instructors. We report the results of this pilot study because it is informative of the initial expectation one can have about the requirements and the results of a study guide. The instructors were asked to look at score sheets that report detailed results of actual students and recommend chapters to study from the course's textbook. This somewhat replicates the context of the computerized study guide.

The setup of this pilot study is listed below:

1.Participants: three seasoned instructors of the course C++;

2.Experimental materials:

(1) Questions and Solutions of an exam for the course C++ (See Appendix1).

(2) Two student score sheets (total scores of 11.5/20 and 7.5/20). Each question on the score sheet is divided into a few sub-items. Each of the sub-items is evaluated.

(3) Table of Contents. (See Appendix 2) We made a table of contents referring to a few books related to C++ programming language and this table of contents includes all the concepts and skills covered by the exam.

(4) Recommendation form. (See Appendix 4) The recommendation form was designed to collect recommendations from the instructors. In the recommendation form, there are two lists of recommendation: "Review" and "Focus". "Review" refers to those concepts or skills that the student has not mastered completely and has to review next. "Focus" refers to those concepts or skills that the student knows nothing about and has to focus on. There is another latent list "mastered" since the chapters that the student doesn't have to focus or review must be those chapters the student has already mastered.

3. Tasks of the participants: to read the questions, solutions and the score sheets of two students before they can give recommendations according to the table of contents.

The pilot study seems simple; however, it didn't proceed well as expected. In retrospect, this was obviously not an easy task for the instructors. One instructor did not finish the recommendation form and made the comment: "The information on the score sheets is not enough for me to make any recommendation to the students." Although the other two instructors filled out the recommendation forms, they both indicated that they didn't feel confident about their recommendations because they couldn't tell what exactly the weaknesses of the students were.

The most interesting finding from this pilot study was that making recommendations by only using the score sheets was not very effective. Therefore, we decided to include not only the score sheets but also the complete answer sheets of the students in the experiment in order to gather effective professional knowledge recommendations. Besides, this finding is important since it may suggest that a computerized study guide could not do as well as an instructor given it cannot analyze detailed answers. Thus, we were motivated to investigate the value of the diagnosis of a detailed answer in the process of making personalized recommendations.

5.3 The design of the experiment

Following the pilot study result, a more thorough experiment was conducted to collect information about professional recommendations for each individual student, to determine the agreement among recommendations from eight instructors, and to investigate the instructors' perceptions regarding the value of detailed answer results in improving personalized recommendations, the potential effectiveness of the proposed system, and the instructors' comments and suggestions for the proposed system. The results from this experiment will serve as data for devising a recommendation module in our proposed intelligent study guide. The following section describes the way the experiment was set up.

Table 2 A profile of the participating instructors

Background of the 8 participating instructors	**Number**	**Percent**
Position:	3	37.5
Professor	1	12.5
Analyst	4	50
Teaching assistant		
Education level:	5	62.5
Ph.D.	2	25
Master	1	12.5
Bachelor		
Major field of study:	3	37.5
Programming/algorithm	3	37.5
Artificial Intelligence	1	12.5
Network Programming	1	12.5
Human computer interaction		
Experience in teaching 1 –2	2	25
years	1	12.5
Or Education 2-5 years	5	62.5
More than 5		

Background of the 8 participating instructors		Number	Percent
years			
Experience in teaching	Never	1	12.5
C++ or algorithm:	1 to 3 terms	1	12.5
	Over 3 terms	6	75
Total participating teachers		8	100.0

5.3.1 Selection of Subjects

The subjects (n=8) (see Table 2) were instructors involved either in teaching the courses of C++, algorithms, other programming languages, or the courses related to programming. Contacts were made by referrals, telephone calls and emails. All the volunteers came from École Polytechnique de Montréal and Concordia University. The instructors were chosen on the basis of having previous experience in teaching a course in C++. The experimenter spent 15 minutes to explain the experiment, administer an interview and demonstrate the procedures for each one of the subjects individually. Then the subjects spent approximately 90 minutes to conduct the experiment.

5.3.2 Experimental Materials and Subject Tasks

The materials used in this experiment are listed below:

1. Table of Contents, exam questions and solutions, and recommendation form. These materials are the same as those used in the pilot study (See section 5.2).
2. Complete answer sheets. For some stages of the experiment (See section 5.3.3 experiment procedure), the actual complete answers from students were provided in this experiment in order to examine the diagnosis value of detailed answer.
3. Questionnaire. A questionnaire was administered at the end of the experiment. The questionnaire contains four sections (See Appendix5). The first section was designed to collect data on the subject's school, education level, major field of study and teaching experience. The second section was designed to investigate the value of detailed information in the process of making recommendations. The third section was designed to examine the potential usefulness of the proposed system. The fourth section was designed to obtain the instructors' suggestions and comments to the proposed system.

At the beginning of the experiment, we administered an interview with instructors, aiming to investigate the instructors' experiences in terms of study plan, study guidance and making recommendations (See Appendix 3 for the interview questions). The subjects were asked to give recommendations to 6 students according to the table of contents and fill out the recommendation form. At the end of the experiment, the instructors were also asked to fill out the questionnaire.

5.3.3 Experimental procedure

Firstly, we estimated the workload for the instructors and decided to include only 6 students' results in the experiment in order to conduct this experiment within two hours. The instructors were asked to give recommendations to those 6 students, who are selected in different levels for this experiment: three have medium performance (M), two have low performance (L), and one has high performance (H).

The major procedure of making recommendations was divided into two steps. In the first step, the instructors were asked to give recommendations only using three score sheets (L=1, M=1, H=1). The 3 score sheets were administered in different order to individual participant in order to make sure all the possibilities of the combinations were covered: (L, M, H); (L, H, M); (M, L, H); (M, H, L); (H, L, M); (H, M, L). Then they were given the complete answer sheets of the same three students to check if it is necessary to modify their recommendations.

In the second step, the instructors were asked to give recommendations using both score sheets and answer sheets of another three students (M=2, L=1). We included more medium scores (M) since they are the least trivial to make recommendations to whereas high scores (H) are usually recommended nothing. The 3 score sheets were also administered in different order to individual participant in order to make sure all the possibilities were covered, (M, L, M), (L, M, M) and (M, M, L).

Detailed experimental procedure:

- Initial contact was made and appointments were established for each subject to participate in the experiment;
- Each participant was asked to sign a consent form;
- At the beginning of the experiment, the experimenter spent 15 minutes to

explain the purpose, the context of the experiment;

- The experimenter spent another 15 minutes to administer an interview to investigate the instructors' experiences about study planning and the methods they preferred to give recommendations. All the interview questions are listed in the "instruction"(see Appendix);
- The author spent 5 minutes to explain the two stages of giving recommendations, explain all the materials available and demonstrate the experimental procedure;
- The subjects started the experiment with the first stage: giving recommendation by using 3 score sheets. After the recommendations were given, the subjects were given the three corresponding complete answer sheets to check if they need to make any modification to their original decision.
- The subjects started the second stage of giving recommendations: using both the score sheets and the complete answer sheets to make recommendations to another three students;
- The subjects were asked to fill out the form "recommendation from teachers";
- The subjects were asked to fill out a questionnaire.

5.4 Analysis of the data

Due to the nature of this empirical study, and in accordance with the majority of the studies of this nature, both quantitative and qualitative approaches were employed for analysis and interpretation of the findings.

5.4.1 Quantitative analysis of the Data

Quantitative analysis of the data in this study is based on Kappa statistic analysis and descriptive analysis of the response to a questionnaire.

The agreement among recommendations from eight instructors can determine what recommendation a study guide should deliver and it can set the desired result of a study guide. We use Kappa statistics (Cohen,1960) to examine the agreement among recommendations from eight instructors since kappa coefficients are measures of correlation between categorical variables often used as reliability or validity coefficients. In this experiment, instructors give recommendations to each student, categorizes the table of contents in the format of "focus" and "review". Those two can be incorporated into one category "non-mastered". There is another latent category "mastered" since the chapters that the student doesn't have to focus or review must be those chapters the student has already mastered. Hence, the agreements on classification tasks among the instructors are assessed in order to verify the degree of agreement among the professional recommendations.

Landis and Koch (Landis and Koch,1977) have characterized different ranges of values for kappa with respect to the degree of agreement they suggest. In this experiment, we take the following standards to evaluate the degree of agreement that the kappa coefficient represents, with $k<=0.2$ considered as slight agreement, $0.2<k<=0.4$ as fair agreement; $0.4<k<= 0.6$ as moderate agreement, $0.6<k<=0.8$ as substantial agreement and $k>0.8$ as almost perfect agreement.

In addition, descriptive analysis of the data was performed by frequency distribution of the responses provided by the participating instructors to certain items of the questionnaires in an attempt to examine the value of diagnosis of detailed answer in making accurate personalized recommendations and to analyze the perceptions of the instructors regarding the potential usefulness of the proposed system.

5.4.1.1 Agreement among recommendations from different instructors

In 5.3.4, we mention that different sequences of the score sheets and answer sheets were administered to different participant in the experiment. The purpose of this arrangement is to make sure that different sequence would not be attributed to the agreement of the recommendations from different instructors.

The recommendation results from instructors are attached in Appendix 6. We apply kappa statistics to calculate the agreement among recommendations from the 8 instructors. The statistic results for the agreement among recommendations to each student from instructors are tabulated in table 3. We also plot the results in figure 6.

As we stated before, the recommendations from instructors can be regarded as three categories: M (master), R (review), and F (focus). If we take review and focus in one category, then the recommendations become two categories, M, R+F. Similarly, the recommendations can also be represented as R, M+F; or F, M+R. In table 3, we calculated four kinds of agreement: K1 (kappa 1) represents the agreement among recommendations given two categories: M, R+F; K2 (kappa 2) represents the agreement among recommendations given two categories: R, M+F; K3 (kappa 3) represents the agreement among recommendations given two categories: F, M+R; K represents the kappa coefficient given three categories: M, R, F.

From Table 3, we can see that the mean value of K3 is over 0.8, which represents perfect agreement. Besides, the mean values of K1 and K are between 0.6 and 0.8, which represents substantial agreement. Only the mean value of K2 is between 0.4

and 0.6, which represents moderate agreement.

From Figure 6, we can see that most kappa scores are above 0.6, which means substantial agreement. Besides, the overall values of K1 and K3 are higher than K2; the overall values of K are in an acceptable level.

Table 3 Kappas for agreement among recommendations from instructors

Student's ID	K1	K2	K3	K
14	0.63	0.35	0.92	0.64
36	0.64	0.64	1	0.64
98	0.85	0.58	0.59	0.63
75	0.77	0.52	0.79	0.71
73	0.85	0.45	0.61	0.64
96	0.97	0.64	0.91	0.90
Mean Kappa	0.78	0.53	0.80	0.69

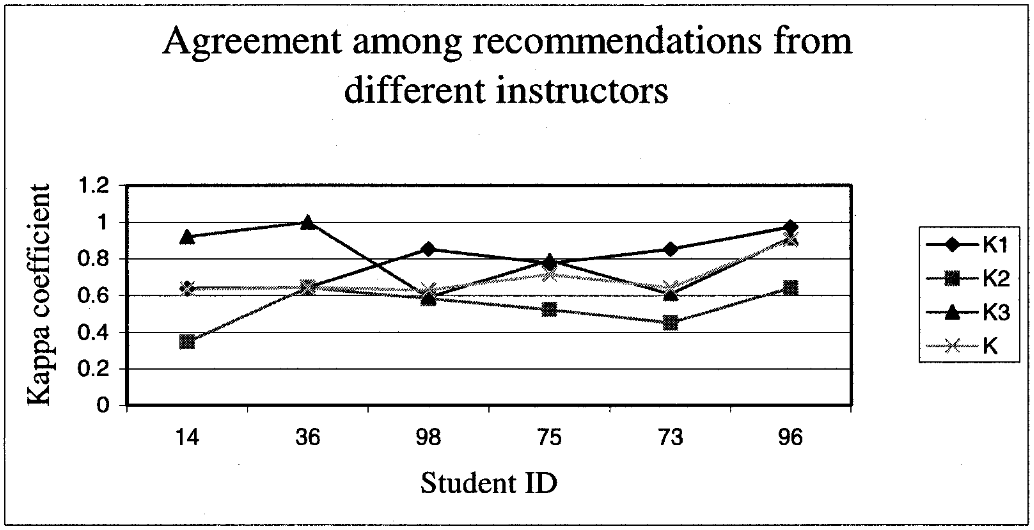


Figure 5 Agreement among recommendations from instructors

The relatively low value of K2 is not surprising because distinction between review and focus is more difficult to make. However, those findings above still indicate that the overall agreement among recommendations from different instructors is acceptable. These results are encouraging and suggest that there exist an agreed upon list of recommendations that a study guide should aim for.

Table 4 Number of recommendation corrections by each instructor

Number of recommendation corrections by each instructor								
Student ID (Performance)	Instructor 1	Instructor 2	Instructor 3	Instructor 4	Instructor 5	Instructor 6	Instructor 7	Instructor 8
36 (13.75/20)	2	3	1	4	1	2	2	2
14 (18.5/20)	1	0	0	2	1	0	0	1
75 (7/20)	3	1	0	1	0	1	1	0

We make an overall recommendation for each student by summarizing all the recommendations from the eight instructors (This result is also reported in Appendix 6). This overall recommendation is regarded as the professional one and will be emulated in the recommendation module in our proposed study guide.

5.4.1.2 Instructors' perceptions of the usefulness of details in helping them to give recommendations

One of the objectives of this experiment is to investigate the value of diagnosing detailed answers in terms of improving recommendations. As we mentioned in 5.3.3, some of the experimental procedure are designed for this purpose.

In appendix 6, we report the detailed information of recommendation modifications made by instructors after evaluation of complete answer sheets. The number of recommendation corrections from each instructor is also summarized in Table 4. We can see that all the instructors made at least one correction in the recommendation form. Besides, instructors made most recommendation corrections for student 36, who is in medium knowledge level (with the score of 13.75/20, see table 4). These findings indicate that the diagnosis of detailed answers assist instructors in making appropriate recommendation, especially for medium level students.

In addition, we attempt to investigate the instructors' perceptions of the usefulness of details in helping them to give recommendations. The data presented in Table 5 summarizes their responses to the questionnaire. The following results can be derived from the frequency distribution of the response provided by the participating instructors to each item of the questionnaire:

1. In the case of making recommendations by only using the score sheets, 75 % of the instructors don't feel confident about their recommendations, 12.5% of the instructors had no opinion, and the remaining 12.5% of the instructors feel confident about their recommendations.
2. In the case of making recommendations by using both the score sheet and answer sheet, 100% of the instructors feel confident about their recommendations.
3. 87.5 % of the instructors indicated that other information about the students, such as attitudes toward study, interest, and previous experience would be also valuable for making recommendations. 12.5% of the instructors had no opinion about the value of other information about the students.
4. 75% of the instructors indicated that a face-to-face question-answering interview

would not be more efficient than the score sheet and answer sheet. 25% of the instructors had no opinion about the value of a face-to-face question-answering interview.

Obviously, the instructors revealed strongly positive perceptions toward the value of the complete answer sheet and the other information about the students such as attitudes toward study, interest, and previous experience. However, most participants do not consider a face-to-face question-answering interview necessary and valuable.

These findings suggest that a computerized study guide may have limitations to emulate human recommendations unless it could analyze detailed answers.

Table 5 Teacher's perceptions of the value of details in helping them to make recommendations

Teachers' perceptions	**Disagree**		**Neutral**		**Agree**	
	N	%	N	%	N	%
I feel confident about my recommendations using only the score sheets.	6	75.0	1	12.5	1	12.5
I feel confident about my recommendations using both score sheets and answer sheets.	0	0.0	0	0.0	8	100.0

Teachers' perceptions	Disagree		Neutral		Agree	
	N	%	N	%	N	%
Other information about the students, such as attitudes toward study, interest, and previous experience would be also valuable for making recommendations.	0	0.0	1	12.5	7	87.5
I feel a face-to-face question answering interview would be more efficient than the score sheet and answer sheet.	6	75.0	2	25.0	0	0.0

5.4.1.3 Instructors' perception of the Effectiveness of the proposed recommendation system

Based on the data presented in Table 5, the following results can be incorporated from the frequency distribution of the responses provided by the participating instructors.

1. All of the instructors demonstrated strongly positive perceptions regarding the effectiveness of the proposed system in improving the students' performance in C++, since 100% of the respondents indicated the system would be useful in improving the students' performance in C++.
2. 100% of the instructors also strongly agree that the system can be used as a teacher's aid in school.

Table 6 Teacher's perceptions regarding the effectiveness of the proposed system

Teachers' perceptions	**Disagree**		**Neutral**		**Agree**	
	N	%	N	%	N	%
This system is useful in improving the students' performance in C++.	0	0	0	0	8	100
This system can be used as a teachers' aid in school.	0	0	0	0	8	100

5.4.2 Qualitative Analysis of the Data

The interview part of this experiment was designed to explore the instructors' former experiences of giving recommendations. Teachers' questionnaire included two open-ended questions designed to explore their opinions regarding the important functions for the proposed recommendation system as well as the instructors' suggestions for the proposed system. Responses to each open-ended question are organized into the following sections.

5.4.2.1 Instructors' Experiences regarding giving students guidance and recommendations

The following is a summary of the experiences regarding giving students guidance and recommendations by the participating instructors.

1. A majority of the instructors had the experience of study plan, study guidance and recommendation.
2. The subjects of the study plan involved Programming, Algorithm, Networking,

Mathematics, Artificial Intelligence, etc.

3. All of the instructors stated that they asked students questions before they could give them recommendations for their study.

   This experience could suggest that an effective assessment driven recommendation module is important in a computerized study guide.

4. The instructors usually asked students 5 to 10 questions in order to diagnose their misconceptions.

   This experience indicates that the use of an adaptive assessment engine will be effective.

5. The process for asking questions usually takes 10-40 minutes, less than 1 hour.

6. All of the participants stated that they asked the oral-based questions. They sometimes even provided guidance by emails.

   This experience implies that the assessment and guidance given by instructors are personalized to each individual student.

7. The study plan was for exams, projects, or homework.

8. The recommendations given by the participating instructors were usually like: read some chapters of a book, do exercises, etc.

9. All of the participants thought the process of study plan were very useful. Some of them got the confirmation by the students' final exam results.

10. Most of the participants indicated that the time a student has to spend on studying what they suggested depended on the student's knowledge level, ability, interest and motivation.

Those experiences listed above could give us some suggestions for devising the recommendation module for our proposed study guide.

5.4.2.2 Instructors' opinions regarding the important functions for the proposed

recommendation system

1. This study guide must have the ability to accurately diagnose the student's misconceptions before it can offer appropriate recommendations.
2. Human-computer interaction and ergonomic issues should be taken into consideration.
3. This system can take advantage of plan or act recognition techniques to compensate the drawback that computers cannot diagnose complete answers from students.
4. This study guide should help students to optimize their study time.
5. This study guide should also model information like student's knowledge background, interest, preferences, or habits in order to guide student's study more efficiently.

5.4.2.3 Instructors' suggestions for the proposed system:

1. This study guide should also provide enough exercises to guide student's knowledge remediation.
2. This study guide should help students to improve their problem-solving abilities.

5.5 Summary

The experiment with instructors presented in this chapter aims to investigate the requirement for the recommendation module of our proposed POKS intelligent study guide. We collect information about professional recommendations for each individual student and investigate the instructors' perceptions regarding the value of details in improving personalized recommendations, the potential effectiveness of the proposed system, and the instructors' comments and suggestions for the proposed system.

A summary of the significant findings from both quantitative and qualitative data is presented below:

1. Kappa statistical analysis revealed that the agreement among recommendations from eight instructors is substantial. Thus, we can take those results as professional recommendations and emulate them in the recommendation module of our proposed study guide.
2. All the instructors made corrections in their recommendation after they evaluate students' complete answer sheets. Especially, instructors made most corrections for students with medium knowledge level. Besides, in their responses to a questionnaire, the instructors revealed strongly positive perceptions toward the value of the complete answer sheet and the other information about the students such as attitudes toward study, interest, and previous experience. However, most participants do not consider a face-to-face question-answering interview necessary and valuable. These findings suggest that computerized study guide may not make as good recommendations as a professional human instructor unless it could analyze the answers. However, we may consider other intelligent techniques such as plan recognition to overcome this drawback.
3. All of the instructors demonstrated strongly positive perceptions regarding the effectiveness of the proposed system in improving the students' performance in C++ and in assisting teaching.
4. The responses to the interview revealed that the instructors' experience of study plan were usually based on personalized assessment and were proved effective. We conclude that the system that can imitate the professional recommendation will have very promising effectiveness.
5. The most important functions of the proposed system perceived by the participants are accurate knowledge assessment, weaknesses diagnosis, Act

prediction for the student, optimizing the use of the study time and interface and ergonomic issues.

6. The most frequently mentioned advice to the proposed system by the participating instructors included: Provide enough exercises and help students to improve their problem-solving abilities.

The experimental results, especially the overall recommendation from the eight instructors, will serve as data for devising a recommendation module in our proposed study guide. More details about emulating these experimental results will be presented in next chapter

CHAPTER 6

RECOMMENDATION MODULE DESIGN AND SIMULATION TEST

6.1 Introduction

In this chapter, we devise a recommendation module that can be integrated with an adaptive assessment engine, such as POKS, for our proposed intelligent study guide by emulating the recommendation results from instructors derived from chapter 5. We implement a simulation test to validate the effectiveness of this recommendation module.

In section 6.2, we present a q-matrix and a corresponding algorithm applied in the recommendation module. The recommendation results from the experiment with instructors serve as data for designing the q-matrix and algorithm. In section 6.3, we make a simulation test to validate the effectiveness of the recommendation module. In section 6.4, we conclude this chapter.

6.2 Devising q-matrix and corresponding algorithm

We adopted q-matrix method to devise the recommendation module in our proposed intelligent study guide. In chapter 4.3.3, we introduced the basic concepts and advantages of q-matrix method. In this section, we will describe how we design the q-matrix and corresponding algorithm.

Firstly, we analyzed the concepts and skills covered by each question item in the exam (for the exam items, see Appendix 1). Then we found the corresponding chapters including those concepts and skills in the table of contents (See Appendix 2, the same one as we used in the experiment with instructors). Then we tried to build the relation between a question item and the chapters in the table of contents. We made a pattern that can reflect the relations between question items and chapters. Finally, a q-matrix is made based on this concept-by-item pattern. The q-matrix we applied in our system is showed in table 7.

We assume the output of an adaptive assessment module is a list of probabilities of success for each test item. This is the case for Bayesian models such as POKS (Desmarais, Maluf et al.,1996). We multiply the probabilities from the adaptive assessment engine by the corresponding values in the q-matrix. Then we calculate the mean value of each column in order to obtain the probabilities of mastery for each chapter. Two threshold values are set: 0.45, and 0.75. That is, when the value of the probability of mastery for a certain chapter is below 0.45, this chapter will be regarded as "focus", which means the student knows nothing about this chapter and has to focus his/her study on it; if the value is between 0.45 and 0.75, then the chapter will be regarded as "review", which means that the student does not completely master this chapter and has to review it; similarly, if the value is over 0.75, which means that the student masters this chapter.

In an optimum situation, the probability list provided by an assessment engine can reflect the true score for each question. To make it simple, we use the full test scores to validate the algorithm.

When comparing the result from our system with the overall recommendation for six students derived from the experiment in chapter 5, we modified the q-matrix and the threshold value a few times until we can get a satisfactory result. Besides, we could also use a classification algorithm to ensure optimal threshold values.

Table 7 Q-matrix

Question	Chapters																		
	1	2.1	2.2.1	2.2.2	2.2.3	2.2.4	2.2.5	2.3	2.4	3.1	3.2.1	3.2.2	2.5	4.2	2.6.1	2.6.2	2.6.3	4.1.1	4.1.2
1	X																		
2.1			X		X														
2.2				X				X	X	X									
3								X		X	X	X	X	X					
4.1		X						X							X	X			
4.2						X		X					X		X	X	X		
5.1							X		X									X	X
5.2				X			X	X	X									X	X

All the programs for realizing the algorithm and the simulation test (section 6.3) are implemented with R.

6.2.1 Preliminary evaluation of our proposed recommendation module

The comparison between recommendation results for six students from the instructors and from the recommendation module are tabulated in Appendix 7. We can see the percentage of agreement between recommendations in Figure 6 and Figure 7.

From the two figures, we can see that most of the agreement is above 80%. So it is a quite substantial agreement. However, the six data cases can only serve as training data to devise the algorithm and they are not enough to confirm the effectiveness of the recommendation module. Therefore, we use a larger sample (51 data cases) as validation data to perform a simulation study in section 6.3.

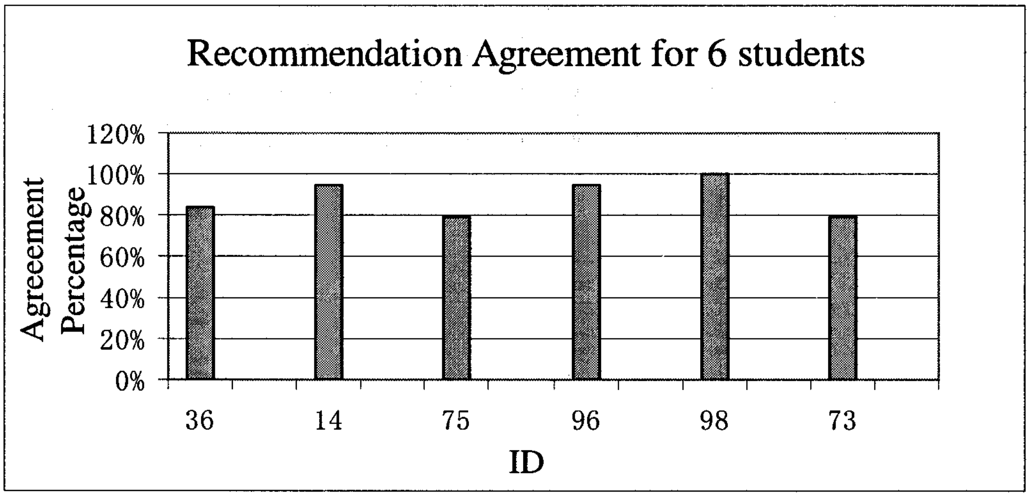


Figure 6 Recommendation agreement for 6 students

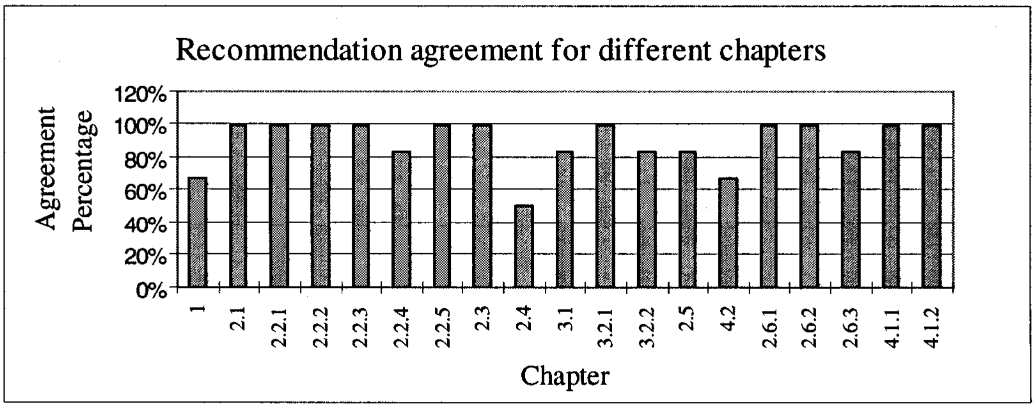


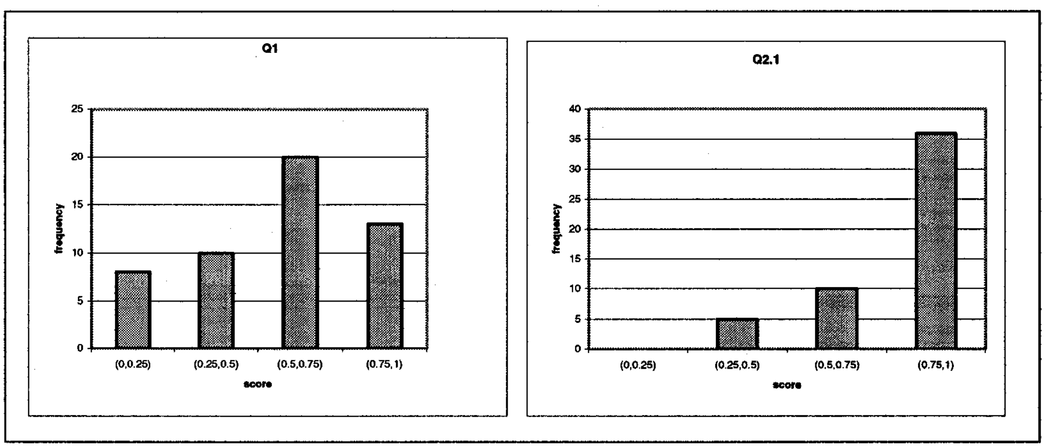
Figure 7 Recommendation agreement for different chapters

6.3 Simulation test

In this section, we implement a simulation test with a larger data sample to validate the effectiveness of the recommendation algorithm introduced in section 6.2.

6.3.1 Test data

All the test data came from the test results of the course C++ in Ecole Polytechnique de Montreal. In this simulation test, we randomly select 51 of the data cases as validation data. The score frequency for each question item is shown in Figure 8. We collect recommendations for the 51 students from our program and a professional C++ instructor separately. Then we compare the two recommendations to examine the effectiveness of the recommendation module. The results of this comparison are analyzed in next section.



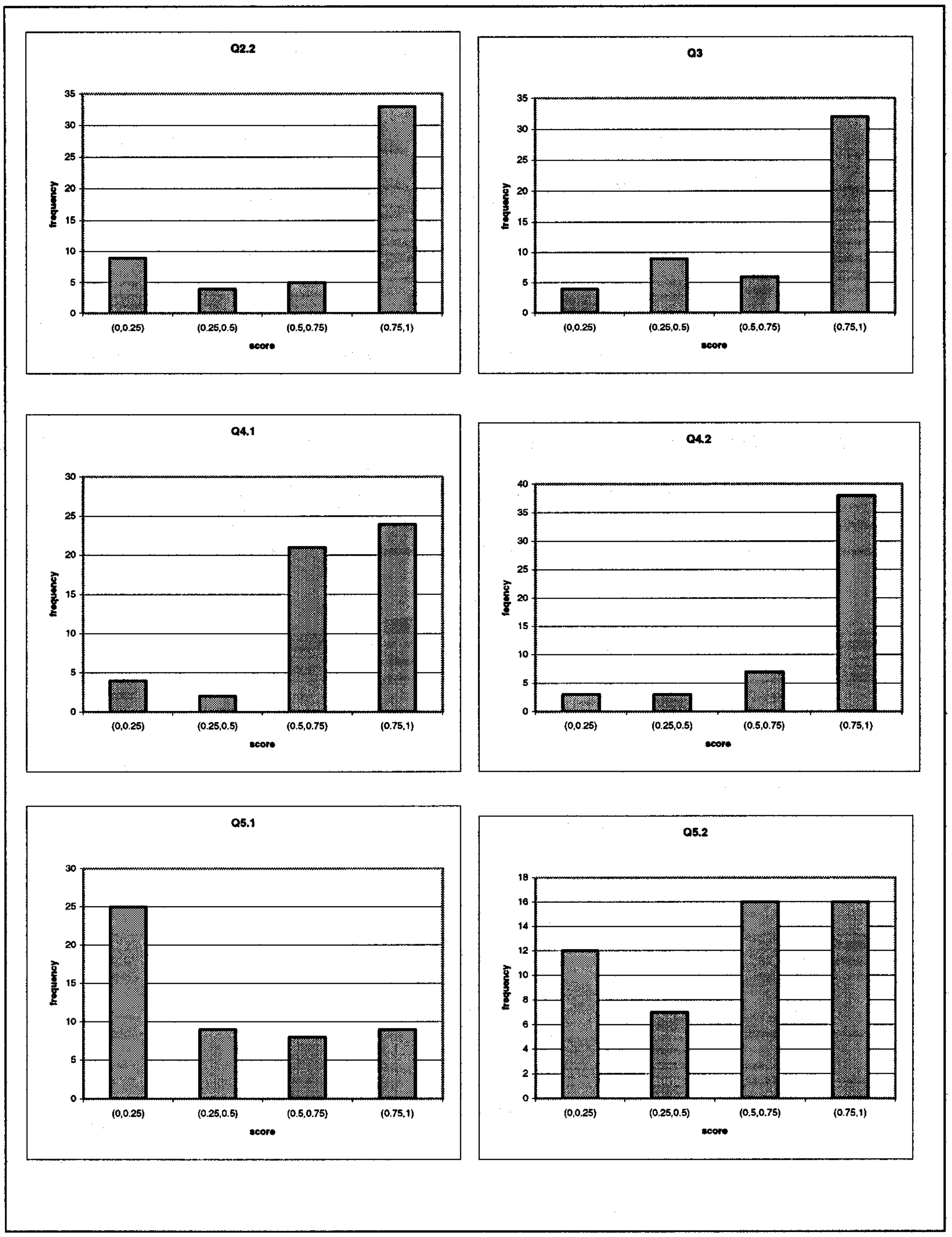


Figure 8 Histograms of score frequency for each question

6.3.2 Simulation results

At the beginning, we collected recommendations for the 51 students from an experienced instructor of C++. The results are considered as expert recommendations. Then we ran our program and compared the results from the program and professional recommendations in order to validate the accuracy of the recommendation module.

The process of making personalized recommendation from our program is described as below:
We set a default value for an initial recommendation, which is based upon the assumption that, at the beginning, all students have the same scores as the average score of the sample data. Thus, this recommendation module does not start with 0% accuracy although the recommendation is not personalized yet. Similarly, an initial value for item prediction accuracy can also be set based on the average scores of the sample students. The question items are sequenced according to the order: most uncertain items first (with an initial probability closest to 0.5) and the most certain ones last (with closest initial probability to 0 or 1). When an item is administered, the score of this item is assigned a true score obtained by a student. The average accuracy of both recommendation and item prediction generated following such an item sequence is shown in Figure 9.

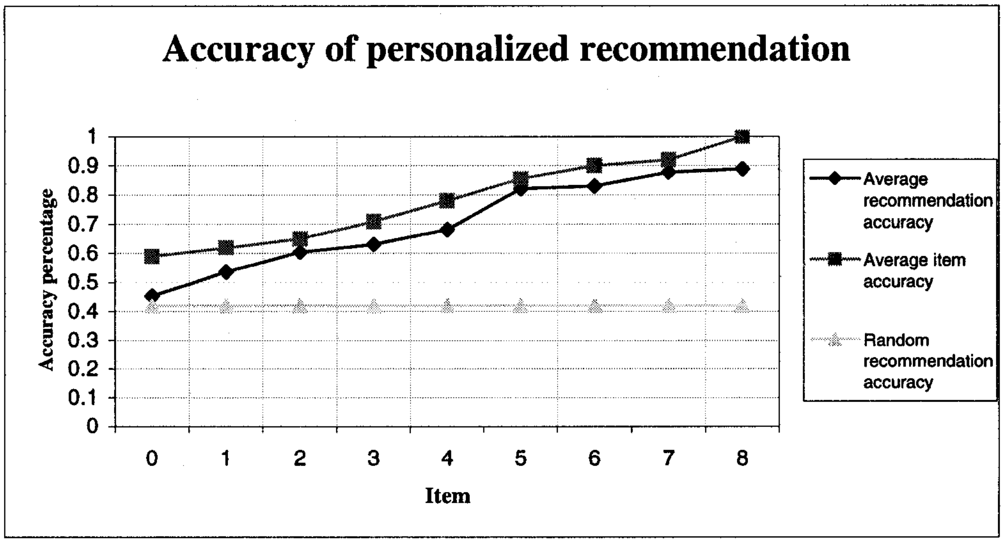


Figure 9 Accuracy of personalized recommendation

In order to have a more thorough analysis of the effectiveness of the recommendation module, we also plot the random recommendation accuracy in Figure 9. We set the program recommendations randomly and make them cover the same percentage for each category (Master, Review, Focus) as the expert recommendation. For example, the "Master" category in the expert recommendation is 35 percent; so we randomly set 35 percent of "Master" in the program recommendations. Then we compare the random recommendation with expert recommendation to yield the random recommendation accuracy.

From Figure 9, we can see that the accuracy of recommendations from our program is superior to random recommendations and it increases gradually when more items are administered. When the responses to all the eight sub-question items are given, the accuracy of this recommendation module reach almost 90%. These findings can basically confirm the effectiveness of this recommendation module.

Besides, the results shown in Figure 9 also indicate that more accurate the item assessment is, more accurate the recommendation is. Thus, the accuracy of recommendation is also dependent upon the effectiveness of the item assessment engine. Undoubtedly, a good item assessment engine, which can efficiently infer accurate performance of a student, is the solid basis for making proper recommendation.

The results of the average recommendation accuracy are basically satisfactory and can also confirm the effectiveness of this recommendation module. However, the agreement of recommendations for certain students or certain chapter or is not as high as other chapters or other students. These findings are not surprising. Firstly, as we concluded in the experiment with instructors in the last chapter, a computerized study guide may not make as good recommendations as a professional human instructor unless it could analyze the answers. Especially, for some students with medium knowledge level, it is hard for a computer to infer their strength and weaknesses only through their score distribution. Moreover, the q-matrix and the corresponding algorithm may need to refine. We can generate optimum weight for q-matrix and more accurate probabilistic inference for the algorithm.

6.4 Summary

In this chapter, we describe the design of a q-matrix and a corresponding algorithm for our proposed intelligent study guide by emulating the recommendation results from instructors derived from chapter 5. Besides, we implement a simulation test to validate the effectiveness of this recommendation module.

From the preliminary evaluation results of the recommendation module, we can see that the comparison between recommendation results for six students from the instructors and from the recommendation module are in a substantial agreement. Furthermore, in order to validate the effectiveness of this module, we implement a simulation test with a large data sample. The results of this test still show that the recommendation from our system is in a good agreement with professional recommendation. However, we still need to further refine the algorithm to improve the effectiveness of the recommendation module.

A summary of the study, general conclusions and tentative proposal for the future research are presented in the next chapter.

Chapter 7

CONCLUSION and FUTURE WORK

This chapter concludes this thesis by providing a summary of the work done and stating our tentative proposals for future work.

7.1 Conclusion

The growing popularity and ease of access to the World Wide Web (WWW) stimulates increasing attention to web-based educational systems. However, since static HTML Web pages are unable to satisfy the heterogeneous needs of many users (Brusilovsky and Maybury,2002), Web-based Adaptive Educational Systems (AES), which aim to increase the functionality of web by making it more personalized for individual learners, have become a hot research area in recent years. A big challenge for web-based educational systems is to provide students with the most suitable pedagogical recommendations that best match their knowledge level.

The work presented in this thesis is a part of a larger ongoing project: the design and implementation of a web-based AES, Poly Intelligent Study Guide, which can guide student knowledge remediation by making personalized assessment-driven recommendation. An inference engine, such as Partial Order Knowledge Structures (POKS), can guide a knowledge recommendation module in this study guide. The objectives of this thesis are to investigate the requirement for this study guide, and to devise an appropriate recommendation module that can be integrated with an adaptive item assessment module.

We implemented an experiment with eight experienced instructors, investigated the process of one-on-one tutoring and collected information about professional knowledge diagnosis and study plan recommendations for each individual student. Both quantitative and qualitative data derived from this experiment were employed to analyze the requirements for the proposed intelligent study guide. Some major findings are briefly summarized here:

1. The agreement among recommendations from eight instructors is substantial. Thus, we can consider those results as expert recommendations and emulate them in the recommendation module of the study guide.
2. All instructors made corrections in their recommendations after they evaluated students' complete answer sheets, especially in the case of making recommendations for students with medium knowledge level. Besides, in their responses to a questionnaire, the instructors revealed strongly positive perceptions toward the value of the complete answer sheet. Thus, a computerized study guide may not make as good recommendations as a professional human instructor unless it could analyze the answers.
3. All of the instructors demonstrated strongly positive perceptions regarding the effectiveness of the proposed system in improving the students' performance in C++ and in assisting teaching.
4. The responses to the interview revealed that the instructors' experience of study plan were usually based on personalized assessment and were proved effective. We conclude that the system that can imitate the professional recommendation will have very promising effectiveness.
5. The most important functions of the proposed system perceived by the participants are accurate knowledge assessment, weaknesses diagnosis, Act prediction for the student, optimizing the use of the study time and interface and ergonomic issues.

6. The most frequently mentioned advice to the proposed system by the participating instructors included: Provide enough exercises and help students to improve their problem-solving abilities.

The experimental results, especially the overall recommendation from the eight instructors, serve as data for devising a recommendation module in the proposed study guide.

We devised a q-matrix and a relatively simple but effective algorithm in the recommendation module that could emulate the overall expert recommendations collected in the experiment. Furthermore, a simulation test was performed to validate the effectiveness of the recommendation module in our intelligent study guide. We compared the recommendations from this intelligent study guide with those from the experts.

The results of this simulation test show that the accuracy of recommendations from our program is superior to random recommendations and it increases gradually when more items are administered. When the responses to all the eight sub-question items are given, the accuracy of this recommendation module reach almost 90%. These findings can basically confirm the effectiveness of this recommendation algorithm. Besides, the results of the simulation test also confirm that more accurate the item assessment is, more accurate the recommendation is. Thus, the accuracy of recommendation is also dependent upon the effectiveness of the item assessment engine. Undoubtedly, a good item assessment engine, which can efficiently infer accurate performance of a student, is the solid basis for making proper recommendation.

The results of the simulation test also indicate that the recommendation accuracy for some certain students or some certain chapter or is not as high as other chapters or other students. These findings are not surprising. Firstly, as we concluded for the last experiment, computerized study guide may not make as good recommendations as a professional human instructor unless it could analyze the answers. Especially, for some students with medium knowledge level, it is hard for a computer to infer their strength and weaknesses only through their score distribution. Besides, the recommendation accuracy may also depend on the type of questions asked. The question types used in our experiment may not be most appropriate as opposed to a standard test such as a multiple-choice question where there are no details to get. Finally, the q-matrix and the corresponding algorithm may need to refine too. We can generate optimum weight for q-matrix and more accurate probabilistic inference for the algorithm.

7.2 Future work

As stated above, our study focuses on investigating the requirement for the recommendation module of an intelligent study guide, and the algorithm we devised is relatively simple but effective. Our work is a part of the design of Poly Intelligent Study Guide. We plan to fully implement this system in the future research. We also can extend and refine our algorithm to a more accurate one that can make more effective assessment-driven recommendations in the future, especially in the case that an adaptive assessment engine is integrated.

In our experiment, the instructors have already indicated that it is difficult to make precise knowledge diagnosis and study recommendations without detailed answers. In fact, the results from the simulation test confirm the relative effectiveness of our recommendation algorithm. Nonetheless, we still need to further investigate the instructors' concerns by other studies with different question types and subjects.

Besides, we can take advantage of data mining technique to automate the process of creating q-matrix. Actually, some researchers already worked on this field, such as the Fault Tolerant Teaching (FTT) system proposed by Barnes (Barnes,2003).

Furthermore, we can take some of the suggestions for the functions of the system from the instructors in the experiment (chapter 5). For example, we should consider adding functions like generating exercises for students and predicting student's actions. These aspects are very important in the instructors' opinion.

Finally, according to the instructors' opinion, human-computer interface design and ergonomic issues should be paid enough attention for implementing this system because these aspects, just like designing the recommendation algorithm and integrating an effective assessment engine, can determine the final success of a certain application.

REFERENCES

ANDERSON, J. R., CONRAD, F. G. et al. (1989). Skill Acquisition and the Lisp Tutor. *Cognitive Science* **13**: 467-505.

BARNES, M. T. (2003). The Q-Matrix Method of Fault-Tolerant Teaching in Knowledge Assessment and Data Mining. Computer Science, North Carolina State University: 174.

BARR, A., BEARD, M. et al. (1976). The Computer as Tutorial Laboratory: The Stanford Bip Project. *International Journal on the Man-Machine Studies* **8(5)**: 567-596.

BENNETT, F. (1999). *Computers as Tutors: Solving the Crisis in Education.* Sarasota, FL, Faben Inc. Publishers.

BIRENBAUM, M., KELLY, A. et al. (1993). Diagnosing Knowledge States in Algebra Using the Rule-Space Model. *Journal for research in Mathematics Education* **24(5)**: 442-459.

BIRENBAUM, M., KELLY, A. et al. (1993). Diagnosing Knowledge States in Algebra Using the Rule-Space Model. *Journal for Research in Mathematics Education* **Vol.24**(No.5): 442-459.

BLOOM, B. (1984). The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring. *Educational Researcher* **13(6)**: 4-16.

BRUSILOVSKY, P. (1996). Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction* **6(2/3)**: 87-129.

BRUSILOVSKY, P. (1999). Adaptive and Intelligent Technologies for Web-Based Education. *Special Issue on Intelligent Systems and Teleteaching 4* **4**: 19-25.

BRUSILOVSKY, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction* **11(1/2)**: 111-127.

BRUSILOVSKY, P. and MAYBURY, M. T. (2002). From Adaptive Hypermedia to

Adaptive Web, Communications of the Acm. *Special Issue on the adaptive Web* **45(5)**: 31-33.

BUCHANAN, B. G AND SORTLIFFE, E. H. (1985). *Rule-Based Expert Systems: The Mycin Experiments of Thestanford Heuristic Programming Project*, Addison-Wesley.

BUTZ, C. J., HUA, S. et al. (2006). A Web-Based Bayesian Intelligent Tutoring System for Computer Programming. *Web Intelligence and Agent Systems: An international journal* **4**: 77-97.

CHEN, C., DUH, L. et al. (2004). A Personalized Courseware Recommendation System Based on Fuzzy Item Response Theory. *Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04).*

COHEN, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and psychological measurement* **20**: 37-46.

CONATI, C., GERTNER, A. et al. (2002). Using Bayesian Networks to Manage Uncertainty in Student Modeling. *User modeling and User-Adapted Interaction* **12(4)**: 371-417.

DESMARAIS, M. C., FU, S. and PU, X.. (2005). Tradeoff Analysis between Knowledge Assessment Approaches. *Proceedings of the12th International Conference on Artificial Intelligence in Education, AEID'2005*, Amsterdam.

DESMARAIS, M. C., MALUF, A. and LIU, J.. (1996). User-Expertise Modeling with Empirically Derived Probabilistic Implication Networks. *User Modeling and User-Adapted Interaction* **5(3-4)**: 283-315.

DESMARAIS, M. C., MESHKINFAM, P., GAGNON, M. (2006). Towards Learned Student Models with Item to Item Knowledge Structures. *User Modeling and User-Adapted Interaction (to appear).*

DESMARAIS, M. C. AND PU, X. (2005a). A Bayesian Student Model without Hidden Nodes and Its Comparison with Item Response Theory. *The International Journal of Artificial Intelligence in Education.*

DESMARAIS, M. C. AND PU, X. (2005b). Computer Adaptive Testing: Comparison of a Probabilistic Approach with Item Response Theory. *UM 2005 User Modeling: The Proceedings of the Tenth International Conference*, Edinburg.

DIETEL, R. J., HERMAN, J. L. et al. (1991). What Does Research Say About Assessment?, NCREL, Oak Brook.

FALMAGNE, J., DOIGNON, J. et al. The Assessment of Knowledge, in Theory and in Practice.

FALMAGNE, J., KOPPEN, M. et al. (1990). Introduction to Knowledge Spaces: How to Build Test and Search Them. *Psychological Review* **97**: 201-224.

GERTNER, A., CONATI, C. et al. (1998). Procedural Help in Andes: Generating Hints Using a Bayesian Network Student Model. *Proceeding of 15th National Conference on Artificial Intelligence*, Madison, Wisconsin.

HENZE, N. AND NEJDL, W. (2001). Adaptation in Open Corpus Hypermedia. *International Journal of Artificial Intelligence in Education* **12**: 325-350.

HOLT, P., DUBS, S. et al. (1994). The State of Student Modeling. *Student Modelling: The key to Individualized Knowledge-Based Instruction*, Berlin: Springer-Verlag.

JOHNSON, W. L. (2001). Pedagogical Agents for Web-Based Learning. *Proceedings of First Asia-Pacific Conference on Web intelligence*, Maebashi City, Japan.

JONHSON, N., RASMUSSEN, S. et al. (1998). Symbiotic Intelligentce: Self-Organizing Knowledge on Distributed Networks, Driven by Human Interaction. *Proceedings of the 6th International conference on Artificial Life.*, MIT Press.

KAMBOURI, M., KOPPEN, M. et al. (1994). Knowledge Assessment: Tapping Human Expertise by the Query Routine. *International Journal of Human-Computer Studies* **40(1)**: 119-151.

KHUWAJA, R., DESMARAIS, M. C. et al. (1996). Intelligent Guide: Combining User Knowledge Assessment with Pedagogical Guidance. *ITS'96: Proceedings of the Third International Conference on Intelligent Tutoring Systems*, London, UK: Springer-Verlag.

KLIR, G. J. AND YUAN, B. (1995). *Fuzzy Sets and Fuzzy Logic: Theory and Applications*, Prentice-Hall.

KOEDINGER, K. R., ANDERSON, J. R. et al. (1997). Intelligent Tutoring Goes to School in the Big City. *International Journal of Artificial Intelligence in Education* **8**: 30-43.

LANDIS, J. R. AND KOCH, G. G. (1977). The Measurement of Observer Agreement for Categorical Data. *Biometrics* **33**: 159-174.

LEE, S., SUNG, C. et al. (2001). An Effective Conversational Agent with User Modeling Based on Bayesian Network. *Proceedings of First Asia-Pacific Conference on Web Intelligence*, Maebashi City, Japan.

LIN, B. AND HSIEH, C. (2001). Web-Based Teaching and Learner Control: A Research Review. *Computers & Education* **37**: 377-386.

LIU, C., ZHENG, L.et al. (2001). Electronic Homework on the Www. *Proceedings of First Asia-Pacific Conference on Web Intelligence*, Maebashi City, Japan.

LORD, F. M. AND NOVICK, M. R. (1968). *Statistical Theories of Mental Test Scores*. Reading, MA, Addison-Wesley.

MITCHELL, P. D. AND GROGONON, P. D. (1993). Modelling Techniques for Tutoring Systems. *Computers and Education* **20(1)**(55-61).

NILSON, N. (1998). *Artificial Intelligence: A New Synthesis*, Morgan Kaufmann.

PEARL, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Net-Works of Plausible Inference*, Morgan Kaufmann.

ROBLES, V., LFARRANAGA, P. et al. (2003). Improvement of Naive Bayes Collaborative Filtering Using Interval Estimation. *Proceedings of 2nd Annual Asia-Pacific Conference on Web Intelligence*, Halifax, Canada.

SCHULZE, K. G., SHELBY, R. N. et al. (2000). Andes: An Intelligent Tutor for Classical Physics. *the Journal of Electronic Publishing, University of Michigan Press, Ann Arbor,* **MI 6(1)**.

SHUTE, V. J. AND GLASER, R. (1990). A Large-Scale Evaluation of an Intelligent Discovery World: Smithtown. *Interactive Learning Environments* **1(1)**: 51-77.

SHUTE, V. J. AND PSOTKA, J. (1996). Intelligent Tutoring Systems: Past, Present, and Future. *Handbook of Research on Educational Communications and Technology, Macmillan*. New York**:** 570-600.

SLEEMAN, D. AND BROWN, J. S. (1982). Introduction: Intelligent Tutoring Systems. *Intelligent Tutoring systems*: 1-10.

STAUFFER, K. (1996). Applications of Student Modeling, Athabasca University.

SYKES, E. R. AND FRANEK, F. (2003). A Prototype for an Intelligent Tutoring System for Students Learning to Program in Java. *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies*, Athens, Greece.

TATSUOKA, K. K. (1983). Rule-Space: An Approach for Dealing with Misconceptions Based on Item Response Theory. *Journal of Educational Measurement* **20**: 34-38.

TATSUOKA, K. K. (1984). Caution Indices Based on Item Response Theory. *Psychometrika* **49**: 95-110.

TATSUOKA, K. K. (1990). Toward an Integration of Item Response Theory and Cognitive Analysis. *Diagnosing monitoring of skill and knowledge acquisition*, Hillsdale, NJ, Lawrence Erlbaum.

TATSUOKA, K. K. (1991). Boolean Algebra Applied to to Determination of Universal Set of Knowledge States. Princeton NJ: Educational Testing Service.

TATSUOKA, K. K. AND LINN, R. L. (1983). Indices for Detecting Unusual Patterns: Links between Two General Approaches and Pothential Applications. *Applied Psychological Measurement* **7**: 81-96.

TATSUOKA, K. K. AND TATSUOKA, M. M. (1992). A Psychometrically Sound Cognitive Diagnostic Model: Effect of Remediation as Empiricla Validity. Princeton, NJ, Educational Testing Service.

UHR, L. (1969). Teaching Machine Programs That Generate Problems as a Function of Interaction with Students. *Proceedings of the 24th National Confernece.*

URBAN-LURAIN, M. (2004). Intelligent Tutoring Systems: An Historic Review in the Context of the Development of Artificial Intelligence and Educational Psychology.

VARADI, F. AND TATSUOKA, K. K. (1989). Buglib. Trenton, New Jersey: Unpublished computer program.

WANG, Y. AND VASSILEVA, J. (2003). Bayesian Network-Based Trust Model. *Proceedings of 2nd Annual Asia-Pacific Conference on Web Intelligence*, Halifax, Canada.

WENGER, E. (1987). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Los Altos, CA, Morgan Kaufmann Publishers, Inc.

WONG, S. K. M. AND BUTZ, C. J. (2001). Constructing the Dependency Structure of a Multi-Agent Probabilistic Network. *IEEE Transactions on Knowledge and Data Engineering* **13(3)**: 395-415.

WONG, S. K. M., BUTZ, C. J. et al. (2000). On the Implication Problem for Probabilistic Conditional Independency. *IEEE Transactions on Systems. Man, and Cybernetics, Part A: Systems and Humans* **30(6)**: 785-805.

APPENDIX

Appendix 1 Sample questions for the exam of the course C++

QUESTION 1 (4 Points)

Sachant que la représentation du nombre 55 en complément à deux sur **huit bits** est $(00110111)_2$,

1.1 (1 point)

donner la représentation de 55 en complément à deux sur **seize bits ;**

1.2 (1 point)

donner la représentation de -55 en complément à deux sur **huit bits.**

QUESTION 4 (4 Points)

Vous serez sûrement surpris de l'apprendre, mais le Père Noël utilise des fichiers binaires pour la gestion des cadeaux à distribuer aux enfants. Le fichier binaire « **Cadeaux-Can.bin** » recense tous les enfants du Canada et, pour chaque enfant, indique si cet enfant a été sage durant l'année. Si oui, le fichier mentionne le cadeau à distribuer à cet enfant sage.

Le fichier « **Cadeaux-Can.bin** » est donc composé d'un nombre préalablement inconnu (plus de 5 millions) d'enregistrements de **type_enfant**. Un enregistrement de **type_enfant** est définie comme suit :

```
struct type_enfant
{
char Nom_Enfant[50]; //Nom et prénom de l'enfant
bool A_Ete_Sage; //Indique si l'enfant a été sage cette
année.
          //(true si l'enfant a été sage, false sinon)
char Cadeau[50]; //Description du cadeau à distribuer à
l'enfant.
          //Chaîne vide si l'enfant n'a pas été sage.
```

```
};
```

4.1 (1 point)

Le père Noël a constaté qu'il avait oublié un enfant dans sa liste et qu'il faut l'ajouter dans le fichier « **Cadeaux-Can.bin** ».

Écrivez la fonction `Ajouter()` qui reçoit en paramètre un enregistrement de `type_enfant` déjà initialisée et qui l'insère simplement à la fin du fichier « **Cadeaux-Can.bin** ». La fonction ne reçoit aucun autre paramètre que l'enregistrement de `type_enfant` et ne retourne aucune valeur au point d'appel.

N'oubliez pas de valider l'ouverture du fichier et faites attention de ne pas écraser le contenu du fichier lors de son ouverture (il serait dommage que le Père Noël perde son fichier de cadeaux à distribuer à plus de 5 millions d'enfants, à cause d'une erreur de votre part).

4.2 (3 points)

Les lutins du Père Noël ont complètement oublié de fabriquer des trains miniatures cette année. Le père Noël a donc choisi de remplacer dans le fichier binaire le cadeau de tous les enfants qui auraient dû recevoir un train par un ours.

Écrivez la fonction **`Remplacer()`** qui ne reçoit aucun paramètre et qui retourne au point d'appel le nombre de remplacements faits dans le fichier « **Cadeaux-Can.bin** ». La fonction **`Remplacer()`** doit remplacer tous les champs **`Cadeau`** contenant la chaîne de caractères « Train » par la chaîne de caractères « Ours ».

N'oubliez pas de valider l'ouverture du fichier. Utilisez les fonctions de la librairie **<cstring>** pour manipuler les chaînes de caractères, à savoir :

- **strcpy(A,B)** qui copie le tableau de caractères B dans le tableau de caractères A.
- **strcmp(A,B)** qui compare les tableaux de caractères A et B et qui retourne 0 si A et B sont identiques.

Appendix 2 Table of Contents for course C++

Chapter 1 Numeric representation

Chapter 2 Basic C++ programming

2.1 Structure of C++ programming

2.2 Define and initialize the data object

2.2.1 fundamental types

2.2.2 derived types (pointer and array)

2.2.3 struct type

2.2.4 function of sizeof

2.2.5 memory allocation

2.3 Writing conditional and loop statement

2.4 How to use array and vectors

2.5 String class (including functions like strcmp and strcpy)

2.6 Writing and Reading files

2.6.1 reading files

2.6.2 writing files

2.6.3 functions about writing and reading files (eg. Seekg/ Tellg)

Chapter 3 Procedural programming

3.1 How to write a function

3.2 Invoking a function

3.2.1 Pass by value

3.2.2 Pass by reference semantics

Chapter 4 Generic programming

4.1 The arithmetic of pointers

4.1.1 The basic pointer

4.1.2 Pointer to pointer

4.2 Using the iostream iterators (cin and cout)

Details for table of contents

Chapter 1 Numeric representation

What's bits and bytes? A 1 or 0 value encoded by the setting of a switch. 00 01 10 11. Most computers operate on information 8 bits, or 16, 32, 64 bits. This collection of bits has been given the odd name of the byte.

Octal and hex numbers. Computer mechanics write binary quantities in a base-8 (octal) or a base-16 (hexadecimal) number format.

Octal, hex, and decimal number format conversion.

Signed and unsigned integers.

Floating-point numbers

Encoding text: ASCII and strings

Chapter 2 Basic C++ programming

2.1. How to write a C++ program.

Main function, a function definition, return type, parameter, operator, header file, namespace, constants.

2.2 Define and initialize the data object.

2.2.1 Fundamental data types, char, int, short int, long int, bool, float, double. Long double. Variables.

Scope of variables. Initialization of variables. Introduction of strings

2.2.2 Derived types: compound data types. introduction Arrays, pointers, array definition, initializing arrays, reference operator(&), dereference operator(*), declaring variables of pointer type, points and arrays, point initialization, character sequences char [].

2.2.3 Data struct. what's data structure. what's data elements, members?

2.2.4 Function of sizeof

2.2.5 Dynamic memory. Operators new and new [], operator delete and delete [], dynamic memory in ANSI-C.

2.3 Writing conditional and loop statement

control structures

conditional structure. If and else

iteration structures loops, while loop, do-while loop, for loop

jump statement, break statement, continue statement, the goto statement, the exit function. The selective structure: switch.

2.4 How to use array and vectors. Accessing the values of an array, multidimensional arrays.

2.5. String class. How to use string class (include <string>), string operations such as < <= > >= ==, string class methods(strcmp, strcpy, at, begin, clear, append, end, empty,find, erase, replace, size, swap)

2.6. Writing and reading files

open a file, ofstream, ifstream, fstream classes, open method(filename, mode), ios::in, ios::out, ios::binary, ios::ate, ios::app, ios::trunc

closing a file,

checking state flags. Bad, fail, eof, good

get and put stream points, tellg, tellp, seekg, seekp

text file, binary files

Charpter 3 procedural programming

3.1. How to write a function.

Declaring functions. Type name(parameter1, parameter2,...) {statement}, return statement, scope of variables, functions with no type.

3.2. Invoking a function

3.2.1 arguments passed by value. This means that when calling a function with parameters, what we have passed to the function were copies of their values.

3.2.2 Arguments passed by reference. When a variable is passed by reference we are not passing a copy of its value, but we are somehow passing the variable itself.

Default values in parameters

overloaded functions

inline functions. It does not change the behavior of a function, but serves to indicate the compiler that the code the function body generates shall be inserted at the point of each call.

Recursivity.

Chapter 4 Generic programming

4.1. The arithmetic of pointers

4.1.1 The basic pointer

4.1.2 Pointers to pointers, void pointers, null pointer, pointers to

structures,

pointers to functions.

4.2 Using the iostream iterators, cin, cout,

Appendix 3 Interview questions

1. Have you ever been involved in the process of study plan? For example, give guidance, suggestions or recommendations to some individual students?

2. What was the subject of the study plan?

3. Did you ask the students some questions before you can give them recommendations for their study? If so, how many?

4. How long did it take for asking questions?

5. Were the questions text-based or oral-based? For example, you asked all the students the same text-based questions, or you just asked the students some improvised questions orally?

6. What was the study plan for? For example, preparing for an exam, or a project, etc?

7. What kind of recommendations did you give the students? For example, review some chapters of a book, or do some exercises?

8. Do you think the study plan was useful?

9. How long did it the students to study based on your recommendations before they came back to you?

Appendix 4 Recommendation from teachers

Name of teacher:___________

	ID	Focus (The chapters or sections that the student knows nothing about)	Review (The chapters or sections that the student hasn't mastered completely)	Estimated time for study (minimum)
Student 1				
Student 2				

Student 3					
Student 4					
Student 5					

Student 6				

Appendix 5 Teacher Questionnaire

Teacher Questionnaire

Date: _________

Section 1:

Name : ______________________________

School:______________________________

Position:_____________________________

Education level:______________________

Major field of Study:__________________

Experience in Teaching or Education (check one)

_____Less than 6 months

_____ 6 months- 1 year

______1-2 years

______ 2-5 years

______More than 5 years

Have you ever taught C++ and algorithms? For how many terms?

Section2:

On a scale from 1 to 5 rate the following statements:

	5=Strongly agree	4=Agree	3=No opinion	2=Disagree	1=Strongly disagree
1. I feel confident about my recommendations only using the score sheets.					
2. I feel confident about my recommendations using both the score sheets and answer sheets.					
3. Other information about the students, such as attitudes toward study, interest, and previous experience would be also valuable for making recommendations.					
4. I feel a face-to-face question answering interview would be more efficient than the score sheet and answer sheet.					

Section 3.

We will propose an intelligent study guide, which can emulate exactly the process of giving recommendations in this experiment. Please give us your perceptions of this system.

On a scale from 1 to 5 rate the following statements:

	5=Strongly agree	4=Agree	3=No opinion	2=Disagree	1=Strongly disagree
1. This system is useful in improving the students' performance in the course C++.					
2. This system can be used as a teachers' aid in school.					

Section 4.

Comments and Suggestions:

Note: if you need more space for any question, please use the back.

1. What do you think are the important functions for the proposed recommendation system?

2. Do you have any suggestions for the proposed recommendation system?

3. Do you have any additional comments for this experiment?

Appendix 6 Results of recommendation from instructors

This appendix includes two forms of recommendation results from the experiment with instructors:

The first form is the original one of recommendation results from the experiment with instructors. We put the recommendations for each student into one single table. The recommendations are given in the format: master (which chapters the student has mastered), focus (which chapters the student has to focus on), review (which chapter the student has to review). 1, 2.1, 2.2.1, 2.2.2, 2.2.3, 2.2.4, 2.2.5, 2.3, 2.4, 3.1, 3.2.1, 3.2.2, 2.5, 4.2, 2.6.1, 2.6.3, 4.1.1, 4.1.2 are the number of chapters corresponding to the table of contents (See Appendix 2); Each line (from 1 to 8) is corresponding to the recommendation given by each instructor (from instructor1 to instructor 8); Last line in each form is the overall recommendation to this student by summarizing 8 recommendations.

In the recommendation tables for student 36, 14, 75, some grids are divided into two parts. The italics in the upper grid represent the recommendation from the recommender when he only reviews the score sheets. We employ this format to demonstrate what modifications the instructors have made for the recommendation after they review the complete answer sheets.

The second form of recommendations is the data that we have treated in order to measure the agreement among recommendations by calculating kappa coefficient.

A. Recommendation results from the experiment with instructors:

Table: Recommendations for student 36

36	1	2.1	2.2.1	2.2.2	2.2.3	2.2.4	2.2.5	2.3	2.4	3.1	3.2.1	3.2.2	2.5	4.2	2.6.1	2.6.2	2.6.3	4.1.1	4.1.2
1	F	M	M	R	M	M	F	R	*F* / R	*R* / M	M	M	R	M	M	R	R	F	F
2	R	M	M	M	M	M	F	M	M	M	M	M	*R* / M	M	M	M	R	*R* / F	*R* / F
3	R	M	M	R	M	M	F	M	M	R	M	*F* / R	M	M	M	M	R	F	F
4	R	M	M	M	M	*R* / M	*R* / F	M	M	M	M	M	M	M	M	M	M	*R* / F	*R* / F
5	*F* / R	M	M	R	M	M	F	M	M	M	M	M	M	M	M	R	R	F	F
6	F	M	M	R	M	M	F	M	*R* / M	M	M	M	*R* / M	M	M	M	R	F	F
7	*F* / R	M	M	R	R	M	F	M	M	M	M	*R* / M	M	M	M	M	M	F	F
8	R	M	M	M	M	*R* / M	*R* / F	M	R	M	M	M	M	M	M	M	R	F	F
Overall	R	M	M	R	M	M	F	M	M	M	M	M	M	M	M	M	R	F	F

Table: Recommendations for student 14

14	1	2.1	2.2.1	2.2.2	2.2.3	2.2.4	2.2.5	2.3	2.4	3.1	3.2.1	3.2.2	2.5	4.2	2.6.1	2.6.2	2.6.3	4.1.1	4.1.2
1	*F* / R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	R	R

14	1	2. 1	2.2. 1	2.2. 2	2.2. 3	2.2. 4	2.2. 5	2. 3	2. 4	3. 1	3.2. 1	3.2. 2	2. 5	4. 2	2.6. 1	2.6. 2	2.6. 3	4.1. 1	4.1. 2
2	R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	R	R
3	R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
4	R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	*F* M	*F* M
5	*F* R	M	M	M	M	M	R	M	M	M	M	M	M	M	M	M	M	M	M
6	R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
7	R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
8	*F* R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M
Overa ll	R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M

Table: Recommendations for student 75

75	1	2. 1	2.2. 1	2.2. 2	2.2. 3	2.2. 4	2.2. 5	2. 3	2. 4	3. 1	3.2. 1	3.2. 2	2. 5	4. 2	2.6. 1	2.6. 2	2.6. 3	4.1. 1	4.1. 2
1	F	F	R	F	*F* R	M	R	M	R	F	R	F	R	M	*F* R	*F* R	R	F	F
2	F	F	R	F	F	M	F	F	R	F	R	R	*F* R	M	F	R	R	F	F
3	F	F	R	F	R	M	F	F	F	F	R	F	R	M	F	F	R	F	F
4	F	F	*R* F	F	R	M	F	F	F	F	R	F	R	R	F	F	R	F	F
5	F	F	R	F	R	M	F	R	F	F	R	F	R	M	F	F	R	F	F
6	F	F	R	R	R	M	F	F	F	F	R	F	*F*	M	F	F	R	F	F

													R						
7	F	F	R	F	F R	M	F	F	F	F	F	F	R	M	F	F	F	F	F
8	F	F	R	F	R	M	F	F	F	F	R	F	R	M	F	F	R	F	F
Overall	F	F	R	F	R	M	F	F	F	F	R	F	R	M	F	F	R	F	F

Table: Recommendations for student 96

96	1	2.1	2.2.1	2.2.2	2.2.3	2.2.4	2.2.5	2.3	2.4	3.1	3.2.1	3.2.2	2.5	4.2	2.6.1	2.6.2	2.6.3	4.1.1	4.1.2
1	R	M	M	F	M	M	F	M	M	M	M	M	M	M	M	M	M	F	F
2	R	M	M	F	R	M	F	M	M	R	M	M	M	M	M	M	M	F	F
3	R	M	M	R	M	M	F	M	M	R	M	R	M	M	M	M	M	F	R
4	R	M	M	F	M	M	M	M	M	R	R	M	M	M	M	M	M	F	F
5	R	M	M	F	M	M	F	M	M	M	M	M	M	M	M	M	M	F	F
6	R	M	M	F	M	M	F	M	M	R	M	M	M	M	M	M	M	F	F
7	R	M	M	F	M	M	M	M	M	R	M	M	M	M	M	M	M	R	F
8	R	M	M	F	M	M	F	M	R	R	M	M	M	M	M	M	M	F	F
Overall	R	M	M	F	M	M	F	M	M	R	M	M	M	M	M	M	M	F	F

Table: Recommendations for student 98

98	1	2.1	2.2.1	2.2.2	2.2.3	2.2.4	2.2.5	2.3	2.4	3.1	3.2.1	3.2.2	2.5	4.2	2.6.1	2.6.2	2.6.3	4.1.1	4.1.2
1	F	M	M	F	M	F	R	F	F	R	R	F	F	F	R	R	F	F	F
2	F	M	M	F	M	F	F	M	F	F	F	F	F	F	R	R	F	R	F
3	R	M	M	F	M	F	F	F	F	F	F	F	F	F	R	F	F	F	F

4	F	F	M	F	M	F	F	F	F	F	F	F	F	F	F	F	F	F	F
5	F	M	M	F	M	F	F	F	F	F	F	F	M	F	R	R	F	F	F
6	F	M	M	F	M	F	F	F	F	F	F	F	F	F	R	R	F	F	F
7	F	M	M	R	M	F	F	F	F	F	F	F	F	F	R	R	F	F	F
8	F	M	M	F	M	F	R	F	F	F	F	F	F	F	R	R	F	F	F
Overa ll	F	M	M	F	M	F	F	F	F	F	F	F	F	F	R	R	F	F	F

Table: Recommendations for student 73

73	1	2. 1	2.2. 1	2.2. 2	2.2. 3	2.2. 4	2.2. 5	2. 3	2. 4	3. 1	3.2. 1	3.2. 2	2. 5	4. 2	2.6. 1	2.6. 2	2.6. 3	4.1. 1	4.1. 2
1	F	M	M	R	M	M	F	M	M	M	F	F	R	M	M	M	M	F	F
2	R	M	M	R	M	M	F	M	M	M	F	F	R	M	M	M	M	F	F
3	F	M	M	R	M	M	F	M	M	M	F	F	R	M	M	M	M	F	F
4	F	M	M	R	M	R	R	M	M	M	F	F	R	M	M	M	M	F	F
5	F	M	M	R	M	M	F	M	M	M	F	F	R	M	M	M	M	F	F
6	F	M	M	R	M	M	F	M	M	M	F	F	R	M	M	M	M	F	F
7	F	M	M	R	M	M	F	M	M	M	F	F	R	M	M	M	M	F	F
8	F	M	M	R	M	M	F	M	M	M	R	F	R	M	M	M	M	F	F
Overa ll	F	M	M	R	M	M	F	M	M	M	F	F	R	M	M	M	M	F	F

B. Recommendations for Calculating Kappa

Eight ratings on each of 19 concepts into one of three categories (M, R, F) for student 36				
Concepts	Categories		$\sum_{j=1}^{3} x_{ij}^2$	

	M	R	F	
1	0	6	2	40
2.1	8	0	0	64
2.2.1	8	0	0	64
2.2.2	3	5	0	34
2.2.3	7	1	0	50
2.2.4	8	0	0	64
2.2.5	0	0	8	64
2.3	7	1	0	50
2.4	6	2	0	40
3.1	7	1	0	50
3.2.1	8	0	0	64
3.2.2	7	1	0	50
2.5	7	1	0	50
4.2	8	0	0	64
2.6.1	8	0	0	64
2.6.2	6	2	0	40
2.6.3	2	6	0	40
4.1.1	0	0	8	64
4.1.2	0	0	8	64
Total	100	26	26	1020

Eight ratings on each of 19 concepts into one of three categories (M, R, F) for student 14				
	Categories			$\sum_{j=1}^{3} x_{ij}^2$
Concepts	M	R	F	
1	0	8	0	64

2.1	8	0	0	64
2.2.1	8	0	0	64
2.2.2	8	0	0	64
2.2.3	8	0	0	64
2.2.4	8	0	0	64
2.2.5	7	1	0	50
2.3	8	0	0	64
2.4	8	0	0	64
3.1	8	0	0	64
3.2.1	8	0	0	64
3.2.2	8	0	0	64
2.5	8	0	0	64
4.2	8	0	0	64
2.6.1	8	0	0	64
2.6.2	8	0	0	64
2.6.3	8	0	0	64
4.1.1	6	2	0	40
4.1.2	6	2	0	40
Total	139	13	0	1154

Eight ratings on each of 19 concepts into one of three categories (M, R, F) for student 75				
Concepts	Categories			$\boxed{\sum_{j=1}^{3} x_{ij}^2}$
	M	R	F	
1	0	0	8	64
2.1	0	0	8	64

2.2.1	0	7	1	50
2.2.2	0	1	7	50
2.2.3	0	7	1	50
2.2.4	8	0	0	64
2.2.5	0	1	7	50
2.3	1	1	6	38
2.4	0	2	6	40
3.1	0	7	1	50
3.2.1	0	7	1	50
3.2.2	0	1	7	50
2.5	0	8	0	64
4.2	7	1	0	50
2.6.1	0	1	7	50
2.6.2	0	2	6	40
2.6.3	0	7	1	50
4.1.1	0	0	8	64
4.1.2	0	0	8	64
Total	139	13	0	1154

Eight ratings on each of 19 concepts into one of three categories (M, R, F) for student 96				
	Categories			$\boxed{\sum_{j=1}^{3} x_{ij}^2}$
Concepts	M	R	F	
1	0	8	0	64
2.1	8	0	0	64
2.2.1	8	0	0	64
2.2.2	0	1	7	50

2.2.3	7	1	0	50
2.2.4	8	0	0	64
2.2.5	2	0	6	40
2.3	8	0	0	64
2.4	7	1	0	50
3.1	2	6	0	40
3.2.1	7	1	0	50
3.2.2	7	1	0	50
2.5	8	0	0	64
4.2	8	0	0	64
2.6.1	8	0	0	64
2.6.2	8	0	0	64
2.6.3	8	0	0	64
4.1.1	0	1	7	50
4.1.2	0	1	7	50
Total	104	21	27	1070

Eight ratings on each of 19 concepts into one of three categories (M, R, F) for student 98				
	Categories			$\boxed{\sum_{j=1}^{3} x_{ij}^2}$
Concepts	M	R	F	
1	0	1	7	50
2.1	7	0	1	50
2.2.1	8	0	0	64
2.2.2	0	1	7	50
2.2.3	8	0	0	64
2.2.4	0	0	8	64

2.2.5	0	2	6	40
2.3	1	0	7	50
2.4	0	0	8	64
3.1	0	1	7	50
3.2.1	0	1	7	50
3.2.2	0	0	8	64
2.5	1	0	7	50
4.2	0	0	8	64
2.6.1	0	7	1	50
2.6.2	0	6	2	40
2.6.3	0	0	8	64
4.1.1	0	1	7	50
4.1.2	0	0	8	64
Total	25	20	107	1042

Eight ratings on each of 19 concepts into one of three categories (M, R, F) for student 73				
Concepts	Categories			$\boxed{\sum_{j=1}^{3} x_{ij}^2}$
	M	R	F	
1	0	1	7	50
2.1	8	0	0	64
2.2.1	8	0	0	64
2.2.2	0	8	0	64
2.2.3	8	0	0	64
2.2.4	7	1	0	50
2.2.5	0	1	7	50
2.3	8	0	0	64

2.4	8	0	0	64
3.1	8	0	0	64
3.2.1	0	1	7	50
3.2.2	0	0	8	64
2.5	0	0	8	64
4.2	8	0	0	64
2.6.1	8	0	0	64
2.6.2	8	0	0	64
2.6.3	8	0	0	64
4.1.1	0	0	8	64
4.1.2	0	0	8	64
Total	87	12	53	1160

Appendix 7 Preliminary evaluation for recommendation module design

The comparison of the recommendation from professionals and our system is tabulated below:

Student ID is in the left-most column;

The list from 1, 2.1..to 4.1.2 denotes all the chapters in the table of contents;

For each student ID, line 1 is the overall recommendation from instructors, and line 2 is the recommendation from our system.

The percentages in the bottom line denote the general agreement between the recommendations for each chapter;

The percentages in the right-most column denote the general agreement between the recommendations for each student;

The underlined items denote the differences between two recommendations.

	1	2.1	2.2.1	2.2.2	2.2.3	2.2.4	2.2.5	2.3	2.4	3.1	3.2.1	3.2.2	2.5	4.2	2.6.1	2.6.2	2.6.3	4.1.1	4.1.2	
36	R	M	M	R	M	M	F	M	M	M	M	M	M	M	M	M	R	F	F	84%
	M	M	M	R	M	M	F	M	F	M	M	M	M	M	M	M	M	F	F	
14	R	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	95%
	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	M	
75	F	F	R	F	R	M	F	F	F	F	R	F	R	M	F	F	R	F	F	79%
	F	F	R	F	R	R	F	F	F	F	R	R	R	R	F	F	R	F	F	
9	R	M	M	F	M	M	F	M	M	R	M	M	M	M	M	M	M	F	F	9

	1	2.1	2.2.1	2.2.2	2.2. 3	2.2 .4	2.2 .5	2.3	2. 4	3. 1	3.2. 1	3.2. 2	2. 5	4. 2	2.6. 1	2.6. 2	2.6. 3	4.1. 1	4.1. 2	
6	R	M	M	F	M	M	F	M	F	R	M	M	M	M	M	M	M	F	F	5 %
9 8	F	M	M	F	M	F	F	F	F	F	F	F	F	F	R	R	F	F	F	1 0 . 0 %
	F	M	M	F	M	F	F	F	F	F	F	F	F	F	R	R	F	F	F	
3	F	M	M	R	M	M	F	M	M	M	F	F	R	M	M	M	M	F	F	7 9 %
	F	M	M	M	M	M	F	M	F	R	F	F	M	F	M	M	M	F	F	
	6 7	100	100	100	100	83.	10 0	10 0	5 0	8 3.	100	83	8 3	6 7	100	100	83	100	100	